



PROCEDIMENTO PARA INTRODUÇÃO DE AGILIDADE EM TESTES DE SOFTWARE

Ciro Grippi Barbosa Lima

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Guilherme Horta Travassos

Rio de Janeiro
Agosto de 2013

PROCEDIMENTO PARA INTRODUÇÃO DE AGILIDADE EM TESTES DE
SOFTWARE

Ciro Grippi Barbosa Lima

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Guilherme Horta Travassos, D.Sc.

Prof^a. Ana Regina Cavalcanti da Rocha, D.Sc.

Prof^a. Renata Mendes de Araujo, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2013

Lima, Ciro Grippi Barbosa

Procedimento para Introdução de Agilidade em Testes de Software / Ciro Grippi Barbosa Lima. Rio de Janeiro: UFRJ/COPPE, 2013.

XIV, 261 p.: il.; 29,7 cm.

Orientador: Guilherme Horta Travassos

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2013.

Referências Bibliográficas: p. 138-147.

1. Processo de Teste. 2. Agilidade em Teste de Software. 3. Projeto de Casos de Testes. 4. Engenharia de Software Experimental. I. Travassos, Guilherme Horta II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

A meus pais Vladimir Ruy Barbosa Lima (In Memoriam) e Conceição Grippi Barbosa
Lima.

À minha esposa Denize, meu filho Guilherme e minha filha Bruna, pelo constante
apoio e compreensão.

Agradecimentos

Em primeiro lugar às forças espirituais que me guiaram até aqui.

À minha mãe, pelos conselhos e o carinho constante.

Ao meu pai, por sua presença amiga em alguns dos momentos mais importantes da minha vida.

À minha irmã Lilian Aboudib Barbosa Lima, pelo seu exemplo como profissional, pesquisadora e ser humano.

Ao meu orientador Prof. Guilherme Horta Travassos, pelo constante estímulo e por sua dedicação à pesquisa e ao ensino.

Às professoras Renata Mendes de Araújo e Ana Regina Cavalcanti da Rocha por participarem da banca examinadora da minha dissertação.

Ao Prof. Joaquim dos Santos e ao Mariano Montoni pelas cartas de recomendação para uma das etapas do processo de seleção no programa de mestrado da COPPE.

Ao Marcelo Santos de Mello pelo constante apoio e otimismo.

Ao José Fortuna Abrantes que realizou o estudo das características de agilidade e práticas ágeis em sua tese de doutorado.

Ao Professor Arilo Claudio Dias Neto por suas contribuições na área de pesquisa sobre processos de testes.

Aos meus amigos e companheiros da COPPE, pelo ótimo ambiente de aprendizado. Foi realmente um privilégio compartilhar ideias, debater e aprender com eles durante estes três anos: Jobson Massolar, Breno França, Paulo Medeiros, Victor Vidigal Ribeiro, Leonardo Mota, Karen Miyuki Nakazato, Taísa Guidini, Verônica Taquette Vaz, Rafael Maiani, Talita Ribeiro, Thiago Souza, Gleison Souza, Luciana Nascimento e, Marco Antonio Pereira Araujo.

A Jose Vilmar Estácio de Souza pela proposição de algumas das idéias sobre testes que auxiliaram muito na elaboração da ferramenta proposta neste trabalho.

Um mestrado em tempo parcial só é possível com o apoio de amigos e companheiros de trabalho. Agradeço a Alexandre Pereira, Julio Cunha, Leonardo Silva e Marco Vilardo pelo apoio nas tarefas do dia-a-dia. Ainda destaco alguns companheiros que me auxiliaram nesta jornada contribuindo para a manutenção de um excelente ambiente de trabalho: Rodrigo Okida, Elvys Marchon, Elizangela Chaves, Ana Reis, Gláucio Santos, Carlos Lima, Claudia Moraes, Aluizio Freitas, Rogerio Lopes, Rodrigo Araújo, Erivaldo Junior, Christiano Sá, Gilmar Mendes, Paulo Barros, Gilberto Trindade, Liliane Gomes, Alexandre Rodrigues, Edgar Araújo e vários

outros que de forma direta ou indireta facilitaram o compartilhamento dos meus esforços entre meu posto de trabalho e a pesquisa.

Agradeço também a William Amorim, Rubens Lima, Fernanda Araújo e Marcello Ribeiro pela compreensão e confiança depositada.

Ao Eduardo Costa Carvalho pelo incentivo e pela oportunidade, ao concordar que uma equipe de sua organização participasse do estudo de caso proposto, e também aos participantes deste estudo: Felipe Melo, Carmem Cristiane de Sena, Greciane Araújo, Leonardo Agrize e Simone Mansour. A Cirley Gonçalves que além de participar do estudo, investiu parte de seu tempo para realização de sugestões e críticas à ferramenta.

Ao Paulo Omar, profissional na indústria de software, área de testes, pelas dicas de livros e textos que me auxiliaram a obter uma visão melhor dos desafios inerentes à profissão de analista de teste e testador.

Ao Eduardo Linhares pelo apoio na etapa de desenvolvimento do *wire-frame* da ferramenta proposta. Ao Neilson Carvalho pela participação na programação de telas da ferramenta proposta. Estendo este agradecimento a Eduardo dos Santos Ferreira por ter apoiado esta empreitada.

Agradeço à Tatiana Gonçalves pela sua garra e Altamir Calixto de Moraes pelo trabalho e dedicação. Vivenciamos dificuldades juntos em um contexto de trabalho em que quase não havia procedimentos mínimos de garantia de qualidade, não poderia deixar de fazer uma referência a vocês.

Ao meu irmão Vlademir Scarponi por sua amizade sincera, apoio e incentivo. Aos meus amigos Rita, Bira, Carol, Andrea Valéria, Sonia, Tatá, Marco Aurélio, Chris, Fernanda e César por todas as risadas nestes últimos anos e pela solidariedade compartilhada em nossa longa caminhada.

A Alexandra de Lima e Luana Ventura pelas sessões de fisioterapia.

Ao pessoal administrativo do PESC, Claudia Prata, Maria Mercedes, Gutierrez, Solange Santos e Sônia Galliano pela atenção.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M. Sc.).

PROCEDIMENTO PARA INTRODUÇÃO DE AGILIDADE EM TESTES DE SOFTWARE

Ciro Grippi Barbosa Lima

Agosto/2013

Orientador: Guilherme Horta Travassos

Programa: Engenharia de Sistemas e Computação

As atividades de teste são fundamentais para assegurar a qualidade do produto de software a ser entregue e mantido para os clientes. Testes devem ser planejados, especificados, projetados, construídos e documentados de forma que seja possível repetir ciclos de execução e aumentar sua capacidade em revelar falhas no software. Todavia, os testes de software devem ser receptivos a mudanças nos requisitos, nas tecnologias e no reescalonamento de prioridades sem deixar de apresentar um custo mínimo para atender a cobertura de acordo com as expectativas de qualidade dos stakeholders.

Neste sentido, esta dissertação apresenta um conjunto de procedimentos e ferramental de apoio aos processos de teste com foco em agilidade, desenvolvido com base em evidência, e que visa facilitar a adaptação rápida às mudanças, promover a cooperação entre os stakeholders e antecipar atividades de planejamento dos testes. Os resultados de sua aplicação em uma organização de software selecionada para uso destes procedimentos apresentaram indícios da sua viabilidade e capacidade de redução de esforço no planejamento e execução dos testes.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

PROCEDURE FOR INTRODUCING AGILITY IN SOFTWARE TESTING

Ciro Grippi Barbosa Lima

August/2013

Advisor: Guilherme Horta Travassos

Department: Computer Science and Systems Engineering

Testing activities are key to ensure the quality of software delivered and maintained for customers. The tests should be planned, specified, designed, constructed and documented to make possible its replication in execution cycles aiming at to increase its ability to reveal software failures. Nevertheless, software testing must have the ability to quickly adapt to changes in requirements, technologies and rescheduling of priorities without impacting the minimal costs and coverage criteria according to the stakeholders quality expectations.

Therefore, this dissertation presents an evidence based set of procedures with tool supporting concerned with test processes focused on agility and aiming at facilitating quick adaptation, promoting cooperation among stakeholders and advancing the planning testing activities. The results of its application into a software organization has indicated its feasibility and ability to reduce the efforts concerned with the planning and execution of testing.

ÍNDICE

1	Introdução.....	1
1.1	Contexto.....	1
1.2	Problema.....	3
1.3	Questões de Pesquisa	6
1.4	Objetivos	7
1.5	Metodologia de Trabalho.....	7
1.6	Organização do Trabalho	8
2	Estudos Preliminares	10
2.1	Introdução	10
2.2	Práticas utilizadas em Métodos Ágeis	12
2.2.1	Uma perspectiva histórica	12
2.2.2	Características e Práticas Ágeis em Processos de Teste.	15
2.2.3	Agilidade em Testes de Software – Um estudo de Campo	17
2.3	Processos de Testes	18
2.3.1	Processo Genérico de Testes (DAVIS, 2000).....	18
2.3.2	Processo de Testes (HASS, 2004).....	21
2.3.3	Processos Verificação e Validação (MPS.BR).....	23
2.3.4	Processo de Testes (Dias Neto, 2006)	25
2.3.5	Norma IEEE-829/2008.....	27
2.3.6	Críticas ao Modelo V	28
2.3.7	Processo de testes selecionado.	31
2.3.8	Definição da linguagem para documentação do Processo	31
2.4	Conclusão	35
3	Características de Agilidade e Processos de Testes – <i>Quasi-Revisão</i> sistemática da literatura.....	37
3.1	Introdução	37
3.1.1	Protocolo de revisão sistemática	38
3.1.2	Execução das buscas.....	41
3.1.3	Análise dos dados	42
3.1.4	Ameaças à Validade.....	45
3.1.5	Perspectivas Futuras	46
3.2	Conclusão	50
4	Procedimento para Introdução de Práticas ágeis em Atividades de Teste de <i>Software</i>	52
4.1	Introdução.....	52
4.2	Procedimento para Inserção de Características de Agilidade.....	53
4.2.1	Etapas da Abordagem.....	54
4.3	Conclusão	62
5	Inserção de Práticas Ágeis em um Processo de Teste.....	64
5.1	Introdução	64
5.2	Etapa 1: Caracterização	65
5.2.1	Contexto da execução.....	65
5.2.2	Documentação dos processos da organização e testes padrão.	66
5.2.3	Identificação das diferenças entre os componentes de cada processo....	72
5.2.4	Avaliação das diferenças encontradas para seleção de modificações nas atividades de teste	73
5.2.5	Seleção de práticas ágeis.....	76
5.3	Etapa 2: Planejamento	85

5.3.1	Documentação das Mudanças Propostas.....	85
5.3.2	Fornecimento de Suporte para adoção das práticas escolhidas	89
5.3.3	Planejamento do Estudo de Caso.....	93
5.4	Etapa 3: Execução	110
5.4.1	Preparação.....	110
5.4.2	Execução do Estudo de Caso.....	112
5.5	Etapa 4: Empacotamento	124
5.5.1	Avaliação e Análise	124
5.5.2	Ameaças a Validade.....	130
5.6	Conclusão.	131
6	Conclusão e Trabalhos Futuros	133
6.1	Considerações Finais	133
6.2	Contribuições da Pesquisa	135
6.3	Limitações	135
6.4	Questões em Aberto.....	136
6.5	Trabalhos Futuros	136
	REFERÊNCIAS BIBLIOGRÁFICAS.....	138
	APÊNDICE A – Métodos e Práticas Ágeis	148
A.1	<i>Empowered Teams, Open workspace</i>	148
A.2	Integração Contínua (XP) e Entregas Frequentes (DSDM).....	154
A.3	Iterações e Incrementos	155
A.4	<i>Sprint Backlog & Product Backlog</i>	155
A.5	<i>Small Releases</i>	156
A.6	<i>Active user involvement & On-site customer</i>	157
A.7	<i>Planning game, Daily scrums & Regular build schedule</i>	157
A.8	<i>Self-managed teams; Empowering teams; Feature teams</i>	159
A.9	<i>Test-driven Development</i>	160
A.10	<i>Refactoring</i>	161
A.11	<i>Baselined requirements</i>	162
A.12	<i>Stakeholder Collaboration</i>	164
A.13	<i>Inspections</i>	164
A.14	<i>Configuration Management</i>	165
A.15	<i>Integrated Testing</i>	166
A.16	<i>Simple Design</i>	166
	APÊNDICE B – Características de Agilidade	167
	APÊNDICE C – Atividades do Processo de Teste Padrão	169
C.1	Atividades do Subprocesso de Planejamento de Testes.	169
C.2	Atividades do Subprocesso de Execução dos Testes.....	170
C.3	Atividades de Monitoramento, Controle e Replanejamento.	170
C.4	Atividade de Encerramento das atividades de Teste	170
	APÊNDICE D – Mapeamento Práticas Ágeis versus Atividades de Teste	171
D.1	Backlog de Produto.	171
D.2	Cliente presente	171
D.3	Design simples	172
D.4	Equipe completa.....	172
D.5	Jogo de planejamento	172
D.6	Metáfora	173
D.7	Liberações frequentes	173
D.8	Reuniões diárias.....	174
D.9	Visibilidade de projeto	174

APÊNDICE E – Planilha p/ Comparação Processo de Teste Padrão e Processo de Teste da Organização.....	175
E.1 Classificação	175
E.2 Atividades.....	176
1. Macro-Atividade Planejar Testes (Processo Padrão).....	176
2. Macro-Atividade Projetar Testes (Processo Padrão)	180
3. Macro-Atividade Especificar Casos de Testes (Processo Padrão)	183
4. Macro-Atividade Definir Procedimentos de Testes (Processo Padrão)	186
5. Macro-Atividade Executar Testes (Processo Padrão).....	188
6. Macro-Atividade Analisar Resultados dos Testes (Processo Padrão)	191
E.3 Artefatos dos Processos.....	193
APÊNDICE F – Atas reuniões.....	197
F.1 Apresentação propostas de mudanças para o SEPG.....	197
F.2 Reunião para análise dos resultados obtidos.....	199
F.2.1 Reunião para avaliação de mudanças a serem incluídas na <i>baseline</i> do processo da organização.	199
APÊNDICE G – Identificação de Cenários e CTs	201
G1 – Tabela de Casos de Uso, Cenários e Casos de Teste	201
APÊNDICE H – Formulários de Avaliação do Estudo de Caso	205
H1 – Formulário de Avaliação das Mudanças no Processo.	205
H2 – Formulário de Avaliação do Editor de Casos de Testes.....	206
APÊNDICE I – Formulários de Consentimento de participação no estudo.....	210
I1 – Formulário de Consentimento dos Participantes	210
APÊNDICE J – Formulários de Caracterização dos Participantes	212
J1 – Formulário de Caracterização dos Participantes	212
APÊNDICE K - EditorCT: Ferramenta para Projeto e Construção de Casos de Testes	215
K.1 Introdução	215
K.2 Editor de Casos de Teste	215
K.2.1 Entidades e Requisitos do Editor de Casos de Teste.....	215
K.2.2 Diretrizes para a Criação de Casos de Testes	227
K.2.3 Definição do Wire-Frame	228
K.2.4 Construção e Testes do Editor.....	231
K.3 Conclusão	258
APÊNDICE L - Relação de Melhorias sugeridas para a ferramenta de Edição	260

ÍNDICE DE FIGURAS

Figura 2-1 Modelo V (adaptado de DAVIS (2000)).....	19
Figura 2-2 Processo Genérico Iterativo Proposto por HASS.....	22
Figura 2-3 Compatibilidade norma IEEE-829/2008 c/ IEEE/EIA Std 12207.0-1996....	27
Figura 2-4 Estratégias de uso do padrão fornecidas pela norma IEEE-829/2008	28
Figura 2-5 <i>Agile Testing Quadrants</i> - Marik.(CRISPIN & GREGORY, 2009).....	30
Figura 2-6 Meta-processo para criação e utilização de um PDS. Adaptado de Reis (2003)	32
Figura 3-1 – Quantidade de ocorrências de cada característica nos artigos	43
Figura 3-2 - Quantidade de ocorrências de cada atividade testes nos artigos	43
Figura 4-1 - Abordagem para Inserção de Características de Agilidade em Processos de Testes.....	55
Figura 4-2 – Procedimentos da Etapa 1 de Caracterização	58
Figura 4-3 - Etapa 2 da Abordagem Proposta – Planejamento	60
Figura 4-4 - Etapa 3 Execução	61
Figura 4-5 - Etapa 4 - Empacotamento	62
Figura 5-1 - Packages de cada Processo documentado.	67
Figura 5-2 - Perfis do Processo de Desenvolvimento da Organização.....	67
Figura 5-3 - <i>Templates</i> do Processo de Desenvolvimento da Organização.	68
Figura 5-4 - <i>WorkProducts</i> do Processo de Desenvolvimento da Organização.....	68
Figura 5-5 <i>Tasks</i> do Processo de Desenvolvimento de Software (PDS) da Organização	68
Figura 5-6 - Descrição e Critérios de entrada/saída das <i>Tasks</i> do PDS da Organização.	69
Figura 5-7- Perfis das <i>Tasks</i> do PDS da Organização.....	69
Figura 5-8 - <i>WorkProducts</i> das <i>Tasks</i> do PDS da Organização.....	69
Figura 5-9 - <i>Capability Patterns</i> utilizados no PDS da Organização.....	70
Figura 5-10 - Atividades do Processo de Desenvolvimento de Software da Organização	70
Figura 5-11 - Macro Atividades do Processo de Testes (DIAS NETO e TRAVASSOS, 2006).	71
Figura 5-12 - Proposta Inclusão Macro Atividade Caracterização Testes na Fase 1... 86	
Figura 5-13 - Atividades de Caracterização Testes e Planejamento Estratégia.	87
Figura 5-14 – Macro Atividades da Fase 2 Análise, Projeto e Construção de Software.	87
Figura 5-15 - Atividades que compõem a Macro Atividade Planejar Testes do Produto.	88
Figura 5-16 – Subatividades de “Projetar Testes”.	88
Figura 5-17 Exemplo de Caso Uso desenvolvido para Release B	113
Figura 5-18 Oportunidade de reuso em CTs com nomes distintos.....	114
Figura 5-19 – Oportunidade de Reuso em CTs com mesmo nome.....	115
Figura 5-20 - Reuso de Procedimentos em CTs com mesmo nome	115
Figura 5-21 – Avaliação normalidade da distribuição de Qtde de Linhas dos CTs produzidos.....	122
Figura 5-22 – Avaliação do nível de complexidade dos Casos de Teste de cada <i>Release</i>	123
Figura K-1 - Diagrama de Classes da Ferramenta de Edição de Casos de Testes... 220	
Figura K-2 - <i>Wire-Frame</i> para Tela de Consulta a Casos de Testes	229
Figura K-3 - <i>Wire Frame</i> da Tela de Edição de Casos de Testes	230
Figura K-4 - Tela de Edição de ACLS (Componentes).....	231
Figura K-5 - Edição de Cenários	231
Figura K-6 - Tela Principal do Editor de Casos de Testes (Perfil Gerente de testes) 232	

Figura K-7 - Tela Principal do Editor de Casos de Testes (Perfil Analista de testes)	233
Figura K-8 - Opções do Menu Desenvolvimento	234
Figura K-9 - Opções do Menu Testes	234
Figura K-10 - Opções do Menu “Execuções”	235
Figura K-11 - Tela Lista de Projetos	236
Figura K-12 - Tela de exibição dos atributos de Projeto e Releases existentes	237
Figura K-13 - Tela de edição de Projeto	237
Figura K-14 - Tela de criação de Novo Projeto	238
Figura K-15 - Lista de Releases do projeto selecionado	238
Figura K-16 - Listas de Requisitos de um Release de um Projeto	239
Figura K-17 - Novo Requisito	239
Figura K-18 - Trocando de Foco de Projeto	240
Figura K-19 - Cadastrando um Caso de Uso a partir da Tela de Lista de Requisitos	240
Figura K-20 - Solicitando tela de cadastramento de Caso de Uso	241
Figura K-21 - Tela criação Novo Caso de Uso	241
Figura K-22 - Solicitando Tela Criação de Novo Cenário	242
Figura K-23 - Solicitando Tela Criação de Novo Caso de Teste	243
Figura K-24 - Tela de Lista de Casos de Teste	243
Figura K-25 - Criação de Novo Caso de Teste	244
Figura K-26 - Tela de Edição de Casos de Testes	244
Figura K-27 - Tela Mapa de Casos de Testes	245
Figura K-28 - Tela de Edição de Caso de Teste e Mapa	245
Figura K-29 - Associando uma pré-condição criada previamente a um Caso de Teste	246
Figura K-30 - Associando um procedimento a um Caso de Teste	246
Figura K-31 - Acessando CT a partir de Tela Lista de Casos de Testes	247
Figura K-32 - Tela de Edição de Caso de Teste	247
Figura K-33 - Rastreabilidade de Requisitos até elementos de Caso de Teste	248
Figura K-34 - Criação de Caso de Teste a partir de Cenário exibido no Mapa de Artefatos de teste	248
Figura K-35 - Rastreamento de Procedimentos a Casos de Testes	249
Figura K-36 - Lista de Componentes disponíveis	250
Figura K-37 - Criação de uma ACL	250
Figura K-38 - Tela de Criação de Nova Pré-Condição	251
Figura K-39 - Associando um componente criado previamente a um passo de pré-condição	252
Figura K-40 - Exibindo conteúdo de Componente a ser associado a pré-condição	252
Figura K-41 - Cadastrando passos de um procedimento	253
Figura K-42 - Cadastrando passos de uma Rotina de Verificação de resultados Esperados	253
Figura K-43 - Associando uma imagem a um resultado esperado	254
Figura K-44 - Exibindo Imagem associada a Resultado Esperado	254
Figura K-45 - Lista de variáveis	255
Figura K-46 - Criação de Nova Variável	255
Figura K-47 - Tela de Lista de Projetos de Casos de Teste	256
Figura K-48 - Lista de Projetos de Casos de Testes	256
Figura K-49 - Lista de Casos de testes	257
Figura K-50 - Registrando execução de Caso de Teste	257
Figura K-51 - Lista de Execuções por Caso de Teste	258

ÍNDICE DE TABELAS

Tabela 2-1 - Técnicas em Engenharia de Software referenciadas em métodos ágeis. Adaptado de Jiang e Eberlein (2009).....	14
Tabela 3-1 Artigos recuperados na execução do protocolo	41
Tabela 3-2 Artigos selecionados após aplicação critérios	41
Tabela 3-3 Distribuição das características de agilidade referenciadas	42
Tabela 3-4 Distribuição das atividades de testes referenciadas.....	43
Tabela 3-5 Distribuição dos artigos por ano de publicação	45
Tabela 4-1 Taxonomia para apoiar comparação entre atividades dos Processos	57
Tabela 5-1 - Atividades existentes no Processo de Testes Padrão.....	74
Tabela 5-2 - Atividades existentes no Processo de Testes da Organização.	75
Tabela 5-3 - Sugestões de mudanças nos artefatos de Testes da Organização.....	75
Tabela 5-4 Sugestões de mudanças nos artefatos de Testes da Organização.....	76
Tabela 5-5 - Oportunidades de estudo de práticas ágeis nas atividades de teste.	77
Tabela 5-6 Práticas <i>Backlog de Produto</i> e Cliente Presente X Atividades Teste.....	78
Tabela 5-7 Prática <i>Design Simple</i> X Atividades Teste.....	78
Tabela 5-8 - Atividades de Testes X Equipe Completa e Jogos de Planejamento.	79
Tabela 5-9 - Atividades de Testes X Metáfora e Liberações Frequentes.	79
Tabela 5-10 - Atividades de Testes X Visibilidade de Projeto e Reuniões Diárias.....	80
Tabela 5-11 - Quantidade de chamados Help-Desk	82
Tabela 5-12 - Caso de Testes Ferramenta de Gestão de Projetos – Versão corrente	83
Tabela 5-13 - Caso de Testes Ferramenta Instalador BI – Versão corrente.....	83
Tabela 5-14 - Caso de Testes CT002A – Versão proposta.....	84
Tabela 5-15 - Caso de Testes CT002B – Versão proposta.....	84
Tabela 5-16 – Eventos programados pelo estudo de caso	105
Tabela 5-17 - Evento 1 – <i>Releases A e B</i> usando versão do Processo anterior às mudanças.....	113
Tabela 5-18 - Atividades do Evento 2	117
Tabela 5-19 Atividades Evento 3	118
Tabela 5-20 Avaliação complexidade e importância para o Negócio dos UCs.....	120
Tabela 5-21 – Níveis de Teste por Caso de Uso.....	120
Tabela 5-22 - Classificação Casos de Testes <i>Releases A e B</i>	123
Tabela 5-23 - Quantidade de CTs dos <i>releases</i> e Esforços associados.....	124
Tabela 5-24 - Taxa de Reúso de Procedimentos em cada <i>Release</i>	125
Tabela 5-25 - Esforço Médio de Projeto de CT por <i>release</i>	126
Tabela 5-26 - Esforço Médio de Projeto de CT por <i>release</i>	127

1 Introdução

Neste capítulo são apresentadas as motivações, o contexto do trabalho, e as questões de pesquisa. São também apresentados os objetivos, a metodologia a ser adotada e a organização deste texto.

1.1 Contexto

Ao iniciar a elaboração de um projeto de software, o gerente responsável planeja um conjunto de atividades de análise, projeto e construção, responsáveis respectivamente pela elicitação das regras de negócio e dos requisitos necessários para o atendimento das necessidades dos *stakeholders* e pela representação destas informações em modelos que visam capturar os aspectos da realidade do contexto observado, quando - através de sucessivas transformações - os requisitos elicidados são compreendidos, representados e transformados, até a obtenção do produto final.

Durante esta jornada, técnicas de verificação e validação podem ser adotadas pela equipe de desenvolvimento e demais *stakeholders* com o objetivo de identificar defeitos nos artefatos gerados – e, na sequência, após a correção do defeito, o artefato é retornado para avaliação, em um esquema de *feedback loop* (MCGREGOR & SYKES, 2001).

Em organizações que apresentam um nível maior de maturidade em seus processos de desenvolvimento de software, estas atividades compreendem: (a) a execução de revisões, seja através da realização de walkthroughs ou inspeções - quando é possível reduzir drasticamente o custo das atividades de detecção e correção de defeitos; (b) a execução das atividades de testes do software – contemplando o planejamento, projeto, implementação e execução - quando é determinado se um atributo ou capacidade de um programa, ou sistema atende aos requisitos (GELPERIN & HETZEL, 1988) e (c) uma avaliação se o software desenvolvido funciona de acordo com as expectativas dos *stakeholders* no ambiente destinado a sua utilização. Resultados publicados na literatura técnica de estudos primários apresentam evidências de que inspeções dos artefatos de projeto podem detectar até 2/3 dos defeitos antes da execução dos testes (CONRADI *et al.*, 1999).

Estudos realizados em projetos na indústria confirmam a aplicação do princípio de Pareto para detecção de defeitos em software, i.e., aproximadamente oitenta por

cento dos defeitos encontrados nos testes estão concentrados em vinte por cento dos módulos do produto de software (GITTENS *et al.*, 2005).

Com o objetivo de apoiar a detecção de defeitos, um conjunto de técnicas de inspeções foi desenvolvido e avaliado com o intuito de apoiar o engenheiro de software nas atividades de verificação (KELLY, 1992). Através da adoção destas técnicas uma quantidade substancial de defeitos são encontrados, porém ainda sobra um percentual a ser eliminado através da execução de atividades de teste.

Para que as atividades de um processo de testes instanciadas em um projeto de software possam ser consideradas eficazes, o produto a ser entregue deve estar em conformidade com os requisitos, e os riscos de falha devem estar em níveis aceitáveis. Isto porque o risco de falha não pode ser totalmente eliminado de um sistema - podendo ser reduzido através de um projeto de testes efetivo (DAVIS, 2000).

A eficiência de um processo de testes de software está diretamente relacionada à sua capacidade de detectar a área do software propensa a maior quantidade de defeitos, utilizando a menor quantidade de recursos para isto.

Sob este ponto de vista, a busca por técnicas e métodos de testes que possam endereçar esta demanda por eficiência é um campo de pesquisa para acadêmicos e para a indústria em geral.

Se um determinado projeto possui como premissa a utilização do software por uma quantidade pequena de usuários e sem perspectiva de carga intensa de dados, com baixa concorrência de acesso, provavelmente não será necessário investir no projeto de uma grande quantidade de casos de testes de desempenho. Para outro projeto que apresente baixa instabilidade de requisitos pode não ser necessário o desenvolvimento de um grupo extenso de suítes de testes de regressão automatizados.

A eficiência neste caso está relacionada à priorização do conjunto de riscos mais adequado às exigências do projeto e também, a definição da extensão de atendimento a estes riscos. Desta forma, torna-se necessário incluir estas atividades na etapa de planejamento e projeto dos testes.

A norma IEEE-829/2008 (IEEE, 2008) cobre o ciclo de vida das atividades de testes e prevê o atendimento a estas questões - ao incluir uma atividade específica de identificação e revisão do nível de integridade em seus processos de fornecimento, aquisição e desenvolvimento (macro atividades concepção, requisitos, *design* e implementação).

O nível de integridade é uma categorização proposta, através da qual o Gerente de Testes deverá avaliar conjuntamente com os *stakeholders* as consequências para todos os envolvidos da ocorrência de eventuais falhas,

determinando assim, a gravidade associada à ocorrência de alguma falha no produto a ser desenvolvido.

Esta norma sugere também uma atividade específica de identificação de riscos nos processos de desenvolvimento (macro atividades concepção, requisitos, *design* e implementação) e operação.

De acordo com a classificação do nível de integridade de cada projeto, e a elicitación dos riscos envolvidos nas etapas de projeto, desenvolvimento e operação do software, determinadas atividades e/ou artefatos poderão ser suprimidos, evitando assim a realização de trabalho desnecessário.

No ambiente de negócios da indústria, esta busca por eficiência nos processos de software está inserida em um contexto no qual ocorrem mudanças frequentes de prioridades, alteração do escopo de projetos e, demanda crescente por qualidade, instigando as equipes de desenvolvimento de software na formulação de estratégias para atender estas solicitações em um curto espaço de tempo.

O desafio de absorver estas mudanças no contexto das atividades de teste demanda capacidade ao gestor em atualizar a cobertura necessária, e os riscos a serem atacados preservando uma relação satisfatória em termos do esforço dos testes dispendido e a qualidade final obtida ao produto.

Neste contexto, a equipe de testes é frequentemente demandada a atualizar casos de testes e procedimentos para permitir a verificação das novas versões de funcionalidades produzidas, decorrentes das mudanças de requisitos e prioridades solicitadas pelos *stakeholders*. E cuidar pela preservação dos critérios e estratégias de teste para o acompanhamento das atividades, para apoiar a decisão entre prosseguir para a próxima etapa do processo ou reiterar a etapa anterior.

1.2 Problema

Dias Neto e Travassos planejaram e executaram uma pesquisa de opinião (*survey*) em organizações localizadas em polo de desenvolvimento de software brasileiro, com o objetivo de observar como estas organizações dão importância e efetivamente aplicam práticas de teste de software (DIAS NETO *et al*, 2006).

Para isto estas práticas foram agrupadas em cinco categorias: (a) Genéricas - abrangendo a realização dos testes de unidade, integração, sistema, aceitação e regressão (b) Organização - constituindo de práticas relacionadas às decisões tomadas pelos gestores responsáveis pelas atividades de teste na organização, como a decisão por re-executar os testes sempre que ocorrerem mudanças no software, a realização de treinamentos sobre testes, a utilização de um processo para sistematizar as atividades de testes, a separação das atividades de testes das atividades de

desenvolvimento, e a existência de um responsável ou equipe alocada para a realização dos testes. (c) Planejamento - abrangendo práticas relacionadas à identificação e utilização dos riscos para projetar e organizar as atividades de testes, à documentação do plano de testes, dos procedimentos, Casos de Testes e resultados esperados, o desenvolvimento de Casos de Testes antes da codificação (d) Controle constituída das práticas de registro do tempo dispendido na realização dos testes, registro das falhas e defeitos encontrados, monitoramento da aderência das atividades realizadas em relação a um processo de testes definido, conservação de registro de log e, avaliação dos documentos quanto à sua qualidade. (e) Medição e análise abrangendo a medição da densidade de defeitos encontrados, dos custos dispendidos, da quantidade de testes realizados, e análise regular da tendência em falhas e defeitos encontrados. (f) Ferramentas – incluindo o uso de ferramentas capazes de gerar os procedimentos e casos de testes, gerenciar os testes, estimar esforço de teste e/ou cronograma, medir cobertura nas fases de teste e rastrear e registrar os resultados de execução dos testes.

Cada participante da pesquisa foi solicitado a responder quais práticas são aplicadas em sua organização e quais são consideradas relevantes pelos profissionais.

Nenhuma das organizações possuía todas as práticas aplicadas, a maioria das práticas de teste (48,5%) foi considerada não aplicada e não importante. As práticas consideradas aplicadas e importantes (42,4%) estão divididas nas categorias Genéricas, Organização, Planejamento e Controle. A carência de atividades de medição e análise reflete os resultados obtidos em relação ao uso e importância dada a estas atividades pelas organizações: todas as práticas desta categoria foram consideradas não importantes e não são utilizadas.

Também foi percebido, a partir da análise dos dados, que empresas de maior porte aplicam mais práticas de teste do que as de médio e pequeno porte e que quase não eram utilizadas ferramentas de apoio aos testes. Na avaliação conjunta do grau de importância deste item, contendo dados de empresas pequenas, médias e grandes, o item ferramentas apresentou um baixo grau de importância em relação a outros itens. E na avaliação em que foram considerados apenas os dados das pequenas e médias empresas, as ferramentas de apoio a gerencia de testes foram consideradas as mais importantes em relação às demais. E as ferramentas geradoras de procedimentos e casos de testes apresentaram o segundo maior percentual.

Segundo o pesquisador, este resultado poderia ser decorrente da ausência de certas atividades nos processos de testes executados pelas organizações, pois a automatização de determinadas tarefas só ocorre depois que esta tarefa está definida

pela organização, e os participantes tem entendimento de sua importância e existam diretrizes que apoiem seu uso.

Uma pesquisa mais recente realizada em 2010 em organizações de software na Europa - contando com a participação de analistas de testes e testadores – teve como objetivo identificar para cada prática de teste as diferenças entre o que é executado nas organizações e, o que os profissionais consideravam mais produtivo. E selecionar situações em que estas diferenças trouxessem grandes obstáculos na consecução das atividades do processo de teste (CAUSEVIC *et al.*, 2010).

Ao final da execução deste estudo, e a realização de avaliações quantitativa e qualitativa foi observado, que os profissionais de testes: (a) consideram inevitáveis algumas mudanças no decorrer do projeto, e que estas podem trazer uma quantidade razoável de dificuldades no seu trabalho do dia-a-dia, (b) consideram relevante que a gerência incentive a interação entre equipe de testes e os *stakeholders*, (c) não consideram extremamente relevante a existência de documentação compreensiva como parte essencial do desenvolvimento de software. O estudo também sugere a realização de investigações futuras sobre *test-first*¹, considerando o interesse em sua utilização pelos profissionais de teste.

Sob estas perspectivas, dotar um processo de testes de características que facilitem a adaptação rápida às mudanças, promova a cooperação entre os *stakeholders*, e antecipe as atividades de planejamento de testes pode trazer ganhos na produtividade da equipe envolvida, e representa um desafio de pesquisa.

Por outro lado, a simples escolha e adoção de um método que siga os métodos ágeis, independente das características do projeto pode não ser a melhor estratégia.

Segundo Boehm e Turner métodos ágeis são capazes de tratar as mudanças através da construção de uma visão dos objetivos e estratégias do projeto, e do compartilhamento de conhecimento tácito entre os vários membros da equipe. Porém não são capazes de escalar suas práticas para grandes e complexos projetos, por não promoverem conformidade para aspectos críticos como especificação de interfaces e arquiteturas de linha de produto (BOEHM *et al.*, 2009).

Considerando a abrangência deste tema e a variedade de aspectos a serem pesquisados, foi realizada uma sequência de estudos primários e secundários no âmbito de uma tese de doutorado, para investigar quais características e práticas são desejáveis em processos de software para dota-los de agilidade, e na sequência avaliar quais destas práticas poderiam ser aplicadas em um processo de testes

¹ *Test-First Development* -. Para cada linha de código a ser incluída é preciso existir teste definido para testá-la.

escolhido *a priori*, sendo esta avaliação realizada por autores dos artigos revisão por pares.

A partir do resultado obtido foi projetado e desenvolvido um guia contendo o mapeamento de atividades de testes versus práticas ágeis, práticas ágeis versus características de agilidade, e um corpo de conhecimento a ser utilizado pelas organizações que desejam embutir estas características em seus processos de testes (ABRANTES, 2012).

Contudo, a utilização de cada uma destas práticas ágeis, nas atividades de teste de uma determinada organização deverá levar em conta características de contexto não apenas do projeto - essas já são consideradas no guia proposto - como também aspectos relacionados às restrições e políticas da organização e à própria forma de implementação.

Neste cenário, o planejamento e execução de um estudo para observar como a inserção de uma determinada prática ágil influencia na execução de uma atividade de testes poderão contribuir para uma melhor compreensão destas práticas e dos efeitos decorrentes.

1.3 Questões de Pesquisa

Considerando a existência *a priori* de um conjunto de características de agilidade e práticas ágeis, recuperadas a partir da realização de estudos primários e secundários realizados no âmbito da indústria e academia (ABRANTES, 2012), as seguintes questões são formuladas:

- Existem relatos de uso de características de agilidade e/ou práticas ágeis em processos de teste? Se existem, quais são estas características? Existem evidências dos resultados obtidos a partir de sua adoção?
- Quais são os procedimentos preliminares necessários para o mapeamento das atividades de testes previstas em um processo de testes padrão com as atividades de uma determinada organização, com o objetivo de facilitar o estabelecimento de um modelo para a inserção de práticas ágeis?
- A partir da seleção de uma ou mais práticas a serem utilizadas em um processo de teste, como se dá a aplicação desta(s) prática(s) em atividades de testes de um processo selecionado?

1.4 Objetivos

A meta principal deste trabalho é realizar um estudo de campo sobre as características e práticas ágeis recuperadas através da execução de um protocolo de *quasi-revisão* sistemática (ABRANTES, 2012), selecionando previamente um subconjunto destas práticas e associando-as a um processo de testes estabelecido e executado em uma organização desenvolvedora de software.

Para isto os seguintes objetivos deverão ser alcançados:

- Realizar revisão de literatura complementar com vistas a avaliar a existência de relatos de uso de características e/ou práticas ágeis em processos de teste, e em caso positivo, caracterizar as evidências dos resultados obtidos a partir de sua adoção.
- Propor um conjunto de procedimentos para apoiar o mapeamento entre as atividades de um processo padrão de testes (DIAS NETO e TRAVASSOS, 2006) e o processo de testes de software da organização que pretende inserir a(s) característica(s) de agilidade.
- Escolher um subconjunto das práticas ágeis que constam no guia proposto (ABRANTES, 2012) a partir do contexto de trabalho da organização e eventuais restrições impostas para avaliação de sua adoção.
- Aplicar a abordagem desenvolvida neste trabalho em uma organização, para realização de um estudo de observação no campo.

1.5 Metodologia de Trabalho

A solução a ser elaborada consistiu em dar continuidade a pesquisa iniciada sobre características de agilidade, direcionando novas buscas na literatura técnica para avaliar a existência de relatos sobre relacionamentos entre as características de agilidade e atividades de processos de teste.

Na etapa de revisão da literatura técnica foram recuperados trabalhos sobre processos de testes para verificar a adequabilidade do uso do processo sugerido (DIAS NETO e TRAVASSOS, 2006) baseado na norma IEEE-829/1998. Isto porque foi necessário avaliar eventuais mudanças realizadas nesta norma em sua versão publicada em 2008.

Uma *quasi-revisão* sistemática foi realizada em busca de novas características de agilidade e de relatos de uso das características já mapeadas em atividades de um processo de teste.

Após a realização desta revisão, e eventual atualização da lista de características e práticas ágeis, foram relacionadas todas as atividades e artefatos do processo de testes escolhido para ser usado como padrão.

Após seleção da organização que já possuía um processo de testes estabelecido, foram listadas suas atividades e artefatos para realizar uma comparação entre cada uma das atividades de ambos os processos. Esta etapa de trabalho teve como objetivo a detecção de *gaps* ou particularidades de implementação.

Neste sentido, dependendo das características do processo da organização selecionada para o estudo, poderá ser sugerida a inclusão de uma determinada atividade no processo de teste corrente. O que a um primeiro momento poderia parecer um contrassenso, desde que neste caso seriam acrescentadas atividades com o objetivo de obter eficiência. Porém não é contraditório com a definição da característica de agilidade *leanness*, na qual são removidas apenas atividades desnecessárias. Sendo em alguns casos necessário acrescentar uma atividade para aumentar a eficiência do processo.

Este contexto obriga a adoção de uma abordagem específica na qual inicialmente deverá ser avaliado se o processo atual apresenta falhas em relação às boas práticas de planejamento e projeto de testes, quando serão sugeridos eventuais ajustes - para só então propor as práticas ágeis que visam embutir as características de agilidade.

Após a definição deste mapeamento, e de posse das práticas ágeis contidas no guia proposto, foram selecionados conjuntamente com o representante da empresa um conjunto de práticas para serem aplicadas e, observadas em uma execução do processo de testes da empresa modificado - acompanhado a partir da execução de um estudo de caso.

1.6 Organização do Trabalho

Este capítulo apresentou o contexto de pesquisa e as motivações para realização desse trabalho, bem como as questões de pesquisa, os objetivos planejados e a metodologia de trabalho. Cada um destes itens será detalhado nos seguintes capítulos:

- **Capítulo 2 – Estudos Preliminares:** apresenta uma revisão inicial da literatura sobre o conceito de agilidade, atividades de testes e características de agilidade em processos de software. Os resultados obtidos nesta revisão

preliminar apoiaram a definição do protocolo de uma *quasi*-revisão sistemática sobre características de agilidade em processos de teste.

- **Capítulo 3 – Características de Agilidade e Processos de Testes – Revisão Sistemática da Literatura:** Neste capítulo são apresentados os elementos do protocolo desta *quasi*-revisão sistemática conduzida com o objetivo de pesquisar relatos da presença de características de agilidade em atividades de teste, e relatos de uso de diretrizes para apoiar a incorporação destas características a estas atividades;
- **Capítulo 4 – Procedimentos para Introdução de Práticas ágeis em Atividades de um Processo de Teste:** apresenta um conjunto de procedimentos a serem executados em campo para composição de uma estratégia de introdução de práticas ágeis em um processo de Teste: (a) Elicitação e Documentação das atividades de Testes do processo de desenvolvimento de software de uma organização, (b) Comparação das atividades de testes da organização com atividades de um processo de testes especializado a partir da norma IEEE-829, bem como análise dos artefatos utilizados (c) Seleção de oportunidades de inserção de práticas ágeis a partir de um corpo de conhecimento definido a priori (ABRANTES, 2012); (d) Proposição de um estudo de caso a partir das observações coletadas nas etapas anteriores.
- **Capítulo 5 – Inserção de Práticas Ágeis em um Processo de Teste:** São apresentados os resultados da execução de cada procedimento desde a caracterização da organização até a etapa de empacotamento dos resultados, incluindo o planejamento e execução de um estudo de Caso com o objetivo de observar a adoção de práticas ágeis em uma organização, as avaliações dos participantes, e as lições aprendidas decorrentes da execução do estudo;
- **Capítulo 6 – Conclusão e Trabalhos Futuros:** são apresentadas as considerações finais da dissertação, contribuições para a pesquisa, limitações, questões em aberto, e perspectivas de trabalhos futuros.

2 Estudos Preliminares

Neste capítulo é apresentado como o conceito de agilidade é entendido no escopo deste trabalho. É também apresentada uma revisão bibliográfica inicial da literatura sobre características de agilidade, práticas ágeis e processos de teste.

2.1 Introdução

Segundo o relatório do ano de 2011 da pesquisa sobre o desempenho de projetos de desenvolvimento de software conduzida pelo *Standish Group (CHAOS Report)*, 37% do total de projetos de software analisados obtiveram sucesso, 21% falharam e 42% apresentaram problemas em sua execução. Sendo considerados: (a) projetos bem-sucedidos - aqueles concluídos dentro do prazo e dentro do orçamento contemplando as características e funções especificadas inicialmente (b) projetos que falharam - aqueles cancelados em algum momento durante o ciclo de desenvolvimento, e (c) concluídos com problemas - aqueles que finalizaram, porém consumindo gastos acima do orçamento original, em data posterior ao prazo estabelecido, e apresentando apenas parte das funções e características originariamente planejadas.

Neste mesmo relatório ao ser comparado o desempenho de projetos cujos responsáveis informaram a adoção de métodos ágeis, com aqueles que utilizaram um ciclo de vida *waterfall* (cascata) foram obtidos os seguintes valores: uma taxa de sucesso de 42% para projetos declarados como ágeis, contra uma taxa de sucesso de apenas 14% dos projetos que informaram o uso do ciclo *waterfall* (SCHWABER e SHUTERLAND, 2012).

Fazendo uma breve retrospectiva em anos anteriores, por exemplo, o relatório publicado no ano de 2010 mostrou uma melhoria de desempenho dos projetos, e a organização destacava dentre as razões para o sucesso obtido: (a) a adoção crescente de métodos ágeis, (b) a desaceleração de novos projetos de ERP e CRM, normalmente projetos que apresentam problemas na sua execução e (c) aumento de projetos de modernização, nos quais a maior parte dos requisitos é inalterada, mudando apenas a base de dados ou a linguagem de *front-end* utilizada (d) redução do número de projetos utilizando o ciclo de vida *waterfall* (STANDISH GROUP, 2010).

Em 2006, os métodos ágeis já apareciam como fator de sucesso na realização de projetos de software (JOHNSON, 2006).

Entretanto, ao avaliar estes resultados se torna difícil mapear quais práticas, métodos, e tecnologias ditas ágeis foram utilizadas em quais tipos de projetos, tamanho, domínio de problema, contexto de utilização, perfil de equipe e outras variáveis de contexto.

Parece que o termo “ágil” é utilizado para descrever um amplo espectro de métodos e *frameworks* para gerenciamento, alguns destes apresentando um conjunto de práticas de engenharia de software originário de trabalhos da academia – como ciclos de iteração curta, envolvimento maior do cliente nas etapas de análise e validação dos requisitos - e de algumas variações destas mesmas práticas pela indústria de software.

Mafera e Travassos (2006) pontuam sobre a recorrente proposição de soluções tecnológicas em engenharia de software, em um quadro de relativa imaturidade, no qual os engenheiros são frequentemente confrontados com questões a respeito do custo de adoção de uma determinada tecnologia, qual o retorno de investimento decorrente, sob quais circunstâncias uma tecnologia deve ser adotada, e qual a tecnologia a investir, dado a existência de várias propostas oferecendo aprimoramento em qualidade e produtividade no desenvolvimento. Sendo que algumas respostas poderiam ser obtidas de forma razoável caso a Engenharia de Software fizesse um uso intenso e sistemático de uma abordagem baseada em evidência.

Com o objetivo de buscar por evidências que comprovem estes relatos de sucesso em agilidade, Dyba e Dyngsor (2008) planejaram e executaram uma revisão sistemática sobre métodos ágeis, para recuperar estudos que pudessem auxiliar na determinação dos benefícios e limitações da adoção de métodos ágeis. Na execução do protocolo desta revisão, após a aplicação dos critérios de inclusão e exclusão foram selecionados trinta e três estudos primários e três estudos secundários. Destes 33 estudos apenas um era sobre SCRUM e vinte e cinco sobre XP e os outros não relacionados a um único método específico.

Em alguns destes estudos observa-se a obtenção de resultados com ganhos nas dimensões de qualidade, custo e prazo. Porém, ao comparar o desempenho dos esforços empenhados para a introdução de alguns métodos, o efeito de algumas implantações foi bastante distinto.

Em um estudo de Svensson e Host (2005) foi reportada dificuldade na adoção do método XP em uma grande organização com uma estrutura razoavelmente complexa, e sugerindo que nestes casos é interessante antes de iniciar a implementação do método, avaliar o processo de software corrente, definir claramente

quais são as modificações a serem realizadas e, divulgar uma terminologia comum para simplificar a comunicação, evitando assim subestimar o esforço necessário para adaptar e introduzir as práticas do XP.

Em contraste, Bahli e Zeid (2005) estudaram a adoção do mesmo método em uma organização canadense, e mesmo a equipe não possuindo experiência no uso do XP consideraram as práticas fáceis de serem assimiladas e utilizadas no desenvolvimento de software. O projeto foi entregue um pouco menos atrasado (50% de atraso contra 60% dos métodos utilizados anteriormente) e com um custo significativamente menor (25% a mais do que o estimado, contra uma média de 50% dos projetos anteriores). Bahli e Zeid afirmam que a adoção deste método foi facilitada por “um alto grau de criação de conhecimento facilitado pelo compartilhamento do aprendizado entre desenvolvedores”.

Ao final desta revisão, Dyba e Dyngsor (2008) registram a necessidade de estabelecimento de uma agenda de pesquisa que contemple a realização de novos estudos, e que favoreçam a obtenção de uma maior força de evidências sobre os benefícios e limitações de uso de métodos e práticas ágeis.

2.2 Práticas utilizadas em Métodos Ágeis

2.2.1 Uma perspectiva histórica

No artigo “*The structure of the IS discipline reconsidered: Implications and reflections from a community of practice perspective*”, é registrada a importância do debate sobre a geração de conhecimento na área dos sistemas de informação, e como se dá sua exposição entre as comunidades de pesquisadores, profissionais da prática e observadores (KLEIN E HIRSCHHEIMB, 2008).

Segundo os autores, a estrutura atual destas comunidades influencia nas suas funções básicas de criação de conhecimento e na comunicação interna e externa deste conhecimento.

Caso não haja reconhecimento no efetivo valor do conhecimento não existe nada a comunicar. Porém sem a comunicação, a criação do conhecimento não pode ser reconhecida e difundida por toda a comunidade. Ambas são necessárias para que seus participantes possam se unir em torno de uma identidade compartilhada e visível.

A estruturação da comunidade visa apoiar a formação de uma agenda de pesquisa no campo. Desde que sem uma identidade reconhecida com suas teorias gerais, práticas e padrões de publicações, é difícil assegurar o desenvolvimento de uma tradição cumulativa de conhecimento para a área e, visualizar as possibilidades de avanço das pesquisas.

Ao analisar como se dá a iteração de indivíduos, Klein e Hirschheimb (2008) afirmam que nem sempre os melhores padrões e produtos são aqueles mais difundidos em uma comunidade, e sim aqueles divulgados pela maioria de seus participantes. Isto explicaria porque em alguns momentos da evolução do conhecimento, décadas são passadas, até que uma boa ideia seja reconhecida como válida. Comentam também que este tipo de retardo não acontece apenas em áreas relacionadas a sistemas de informação, mas também a varias outras áreas do conhecimento humano como, por exemplo, a Física (ZUKAV, 2001).

Como forma de endereçar algumas das questões elencadas, os autores sugerem:

- Uma atualização contínua da história evolutiva do conhecimento na área,
- O incentivo de especialização acadêmica (programas de pós-graduação) para o pessoal envolvido (indústria, profissionais da pratica) e,
- A busca por meios que facilitem a comunicação e entendimento dos paradigmas debatidos entre os participantes das redes de comunidades de prática.

Considerando a relevância de atualização histórica, e a oportunidade de contribuir para esclarecer e evidenciar algumas semelhanças, características, e diferenças entre métodos ágeis e métodos ditos “tradicionais”, Jiang e Eberlein (2009) realizaram um mapeamento comparativo entre estas duas abordagens, quando foi ratificado que alguns nomes e terminologias utilizados em metodologias ágeis por vezes traduzem um mesmo conceito já abordado anteriormente em trabalhos sobre técnicas em Engenharia de Software.

Sob a mesma perspectiva Conboy (2009) faz referência a um artigo de Keen (1991) sinalizando que muitos dos conceitos e áreas de interesse de pesquisa no desenvolvimento de sistemas de informação não são questões novas. E frequentemente oriundas de trabalhos realizados anteriormente, frutos de investigações executadas no passado. Esta tendência parece que permanece em relação às áreas do conhecimento em que os métodos ditos ágeis estão inseridos.

Na pesquisa realizada por Jiang e Eberlein (2009), na fase inicial de levantamento foram selecionados e analisados mais de 100 livros e artigos sobre metodologias de engenharia de software publicadas na IEEE SE, ICSE, IEEE e *ACM Transactions on SE and Methodology*. Sendo a pesquisa concentrada na análise de algumas práticas típicas, princípios e contexto tecnológico em que as metodologias relacionadas foram aplicadas.

Este mapeamento, exibido na tabela 2.1, serviu como um guia inicial, um ponto de partida nesta primeira etapa de revisão bibliográfica sobre práticas ágeis. E os comentários subsequentes neste tópico, bem como as descrições de cada prática dispostas no Apêndice A – nem todas extraídas do trabalho de Jiang e Eberlein - registram os desdobramentos de pesquisa da dissertação.

Contudo não foi objetivo nesta etapa cobrir todo o conjunto de metodologias ágeis, nem tampouco realizar uma análise comparativa de seus objetivos e suas especificidades.

Tabela 2-1 - Técnicas em Engenharia de Software referenciadas em métodos ágeis. Adaptado de Jiang e Eberlein (2009)

Práticas em Métodos Ágeis	Referências em ES e outras disciplinas
<i>Empowered teams [DSDM, Lean Development]</i>	(WEINBERG, 1971)
<i>Open workspace [XP]</i>	
Integração Contínua [XP]; Entregas Frequentes [DSDM]	(GILB, 1985)
Iterações e Incrementos [<i>Scrum</i>] [DSDM] [XP]	(LARMAN & BASILI, 2003)
<i>Sprint backlog, Product Backlog [Scrum]</i>	(AOYAMA, 1998)
<i>Small releases [XP]</i>	(GILB, 1988)
Active user involvement (DSDM) ;On-site customer [XP]	(ALEXANDER, 1979)
Planning game [XP]; Daily scrums [Scrum]; Regular build schedule [FDD]	(TAKEUCHI & NONAKA, 1986)
<i>Self-managed teams [Scrum]; Empowering teams [DSDM]; Feature teams [FDD]</i>	(DILL & PEARSON, 1991)
<i>Test-driven Development [XP]</i>	(THOMKE, 1997)
<i>Refactoring [XP]</i>	(DIJKSTRA, 1972)
<i>Baselined requirements [DSDM]</i>	(BELL & THAYER, 1976)
<i>Stakeholder collaboration [DSDM]</i>	(WOOD & SILVER, 1989)
<i>Inspections [FDD]</i>	(FAGAN, 1976)
<i>Configuration management [FDD]; Reversible changes [DSDM]</i>	(BOEHM, 1976)
<i>Integrated testing [DSDM]</i>	(SOMMERVILLE, 2006)
<i>Simple design [XP] Fitness (DSDM)</i>	(BOURQUE, 2002) (SAINT-EXUPERY, 1954)

Isto porque não é a diversidade de métodos, variantes e derivados que dificulta um mapeamento entre as várias práticas ágeis existentes, mas o fato de muitos deles pretenderem abranger atividades distintas de um processo de desenvolvimento, e por vezes com abordagens bastante diversas.

Alguns destes métodos são constituídos por um conjunto de instruções operacionais prescritas para as equipes de desenvolvimento como é o caso do XP (BECK, 1999). Outros fornecem *frameworks* para apoiar as atividades de

gerenciamento de projetos (SCRUM), ou ainda são melhores descritos como um conjunto de princípios filosóficos (POPPENDIECK e POPPENDIECK, 2003).

Existem casos de divergências e orientações contraditórias, por exemplo, XP requer posse coletiva do código (BECK, 1999), já o método FDD (*Feature Driven Development*) recomenda que o código possua uma pessoa responsável, sugerindo propriedade individual (GOYAL e SCHILLER, 2007).

Conboy (2009) comenta que é razoável e talvez inevitável que criadores de métodos diferentes possuam idéias distintas sobre como a agilidade pode ser alcançada. E neste sentido, fornecer uma variedade de opções para as equipes de desenvolvimento poderia ser benéfico para estas equipes. No entanto, isto pode ser muito desafiador e confuso para equipes que desejam ser ágeis – quando os conselhos fornecidos são diametralmente opostos.

As descrições de cada prática disposta na tabela 2-1 e os comentários sobre os artigos relacionados estão dispostos no Apêndice A.

2.2.2 Características e Práticas Ágeis em Processos de Teste.

Considerando a importância de identificar e avaliar características e práticas ágeis Abrantes (2012) planejou e executou duas *quasi-revisões* sistemáticas, para determinar quais são as propriedades e características ágeis utilizadas em processos de *software*, e quais são as práticas de *software* que podem ser consideradas práticas ágeis.

Neste trabalho são feitas duas referências para contextualizar o uso do termo processo e prática: (a) Um processo deve incluir definições para ciclo de vida de desenvolvimento, papéis, atividades, linguagem de modelagem, artefatos, práticas ou técnicas, regras e as chamadas atividades guarda-chuva (TAROMIRAD e RAMSIN, 2009). (b) Práticas são atividades que implementam os princípios que regem os processos ou métodos. Estes por sua vez, são idéias, entendimentos ou objetivos que estão por traz das práticas (JIANG e ARMIN, 2009).

A partir dos resultados obtidos pela execução das duas revisões, foi realizada uma pesquisa de opinião (*Survey*), quando um conjunto de especialistas foi convidado a analisar a relação de características e práticas resultantes da execução dos estudos secundários, para avaliar sua pertinência e ordem de relevância (ABRANTES, 2012).

Após o estabelecimento de uma relação de dezessete práticas e dezoito características o trabalho foi concentrado na avaliação de quais práticas poderiam ajudar uma determinada atividade a alcançar uma determinada característica de agilidade.

Estes estudos subsidiaram o planejamento e execução de uma segunda etapa, na qual foi desenvolvido um corpo de conhecimento disposto através de um guia, com o objetivo de apoiar os gestores e gerentes de projetos de desenvolvimento de software na utilização destas práticas e características.

Na definição do escopo de quais atividades de um ciclo de vida de desenvolvimento seriam elencadas para fazerem parte do guia, foram selecionadas as atividades de testes - considerando as demandas constantes de atendimento à níveis de qualidade do produto a ser entregue, e os custos envolvidos na preparação e execução de testes de software em geral.

Na escolha de um processo de testes contendo as atividades devidamente modeladas e documentadas foi realizada uma revisão na literatura sobre processos desta natureza, e selecionado um modelo de processo baseado na norma IEEE-829 (DIAS NETO e TRAVASSOS, 2006).

No estabelecimento do corpo de conhecimento para apoiar a equipe de testes na adoção destas práticas e características foram utilizados elementos do *framework* ASSF (*Agile Software Solution Framework*) proposto por Qumer e Henderson-Sellers (2008). E ao final deste trabalho foram geradas duas matrizes: (a) Matriz de associação entre características de agilidade e práticas; e (b) Matriz de associação entre práticas ágeis e Atividades de Teste.

Após os mapeamentos estabelecidos, uma avaliação externa foi realizada através do planejamento e execução de uma revisão por pares. E, após esta avaliação, o conjunto de características e práticas ágeis e os mapeamentos avaliados foram carregadas em uma base de conhecimento.

Para sua utilização, o gestor de testes informa as características de seu projeto, e como resultado o guia retorna um grau de adequação à adoção das práticas ágeis. Dependendo deste resultado, o gestor opta por dar continuidade ao uso do guia ou não. Caso seja escolhido pelo prosseguimento, o gestor deverá selecionar as características de agilidade a serem utilizadas, e após esta seleção será fornecido pelo guia um conjunto das práticas associadas às características, as quais poderão ser utilizadas nas atividades de teste, quando da instanciação do processo.

Examinando o relacionamento entre práticas ágeis e atividades de testes – resultado da execução da revisão por pares - percebe-se que algumas destas práticas, utilizadas no âmbito das atividades de planejamento, projeto e construção de software podem ser transpostas e traduzidas para uso em atividades de testes, através de algumas inferências, enquanto outras requerem uma elaboração preliminar e análise mais detalhada.

Contudo, dependendo das premissas utilizadas e a forma como esta inferência será realizada, alguns enganos podem ser cometidos.

Por exemplo - na etapa de planejamento das atividades de análise, projeto, construção e testes das funcionalidades - a utilização da prática de jogos de planejamento para apoiar a priorização dos requisitos do sistema a ser construído é utilizada correntemente por equipes SCRUM com o cliente, quando são considerados os esforços associados e o valor potencial a ser agregado para o negócio de cada funcionalidade. Correntemente esta prática é utilizada para priorização do desenvolvimento de funcionalidades considerando toda a etapa do ciclo de vida de desenvolvimento.

Ao ser utilizado apenas no âmbito de atividades de teste, sua transposição e uso em atividades de planejamento de testes deve considerar este novo contexto.

Uma derivação direta nas atividades de planejamento dos testes pode ser realizada na negociação entre uma fábrica de testes e a empresa desenvolvedora responsável pela entrega do projeto, ou ainda por um setor de *Quality Assurance* (QA) que atende a várias equipes de desenvolvimento.

Todavia quando os testadores junto com os desenvolvedores pertencem a uma pequena equipe, talvez não faça muito sentido utilizar esta prática exclusivamente para o planejamento das atividades de teste, já que seu uso para priorização das tarefas será realizado na etapa de planejamento do projeto como um todo, com a participação de todos envolvidos, o pessoal do negócio e também a equipe de testes.

Por outro lado, considerando a abrangência da pesquisa e o resultado obtido a partir desta revisão por pares, executada por especialistas em testes e pesquisadores, os relacionamentos obtidos podem vir a apoiar efetivamente a inserção de agilidade em atividades de um processo de teste.

As 18 características relacionadas estão dispostas no Apêndice B. As definições de cada uma das atividades de testes de um processo de teste a ser utilizado como padrão estão listadas no Apêndice C. E o significado de cada uma das práticas e seu relacionamento com as atividades de testes estão dispostas no Apêndice D.

2.2.3 Agilidade em Testes de Software – Um estudo de Campo

Realizando uma comparação entre o conjunto das dezoito características e as dezessete práticas ágeis selecionadas por Abrantes (2012) e aquelas recuperadas pela revisão inicial da literatura, percebe-se que estes conjuntos de práticas e características abrangem os aspectos mapeados nesta primeira busca realizada a partir dos trabalhos referenciados no artigo de Jiang e Eberlein (2009).

Como a proposta do trabalho de Abrantes (2012) foi desenvolver um corpo de conhecimento contendo os mapeamentos de práticas e características de agilidade para apoiar sua adoção em atividades de teste; na fase de criação das strings de busca por estas características e práticas, a população foi composta por termos capazes de abranger todos os processos de software, de forma a obter maior sensibilidade (DIESTE & PADUA, 2007).

Dessa forma, o estabelecimento de um novo protocolo de quasi-revisão sistemática, tendo como objetivo buscar por relatos de adoção de características de agilidade, exclusivamente em atividades de teste, pode contribuir para agregar novas evidências e novas perspectivas para o estudo realizado.

A escolha da busca a partir das características e não por práticas foi motivada pelo entendimento de por estarem associadas a um nível maior de abstração, ao procurar por cada uma das características, por decorrência serão recuperados relatos de uso de práticas associadas.

As informações obtidas nesta quasi-revisão visam apoiar e confirmar a pertinência de encaminhar um estudo de campo sobre a adoção de práticas ágeis em um processo de testes formalizado, em uma determinada organização selecionada para este fim.

Antes, porém foi realizada uma revisão na literatura sobre processos de testes em geral, e sobre linguagens para modelagem de processos de software.

2.3 Processos de Testes

2.3.1 Processo Genérico de Testes (DAVIS, 2000)

Ao definir as quatro fases de um processo genérico de testes, Davis propõe um esquema que visa responder quatro perguntas relativas às atividades de testes (DAVIS, 2000):

- O que testar?
- Como testar?
- Quando testar?
- Quais são os resultados?

A fase de análise visa responder à primeira pergunta (o que testar?). Nesta etapa é estabelecido o escopo do teste, através da criação de uma hierarquia das funcionalidades, e as características a serem testadas. Quando deverá ser identificada a importância relativa das condições de Testes e sua rastreabilidade à documentação das funções a serem testadas (em um esquema de *cross-reference*).

A fase de *Design* (Projeto) visa responder à segunda pergunta (como testar?). Nesta fase são analisados e documentados os estímulos (entradas e pré-condições) e os resultados esperados a serem utilizados para demonstrar que as condições de testes foram atendidas. O objetivo é ganhar o máximo de cobertura de Testes com uma quantidade mínima de dados de teste.

Na fase de cronograma (quando testar?) é definida a sequência em que os procedimentos de testes deverão ser executados.

Na fase de execução e avaliação (quais são os resultados?) devem ser monitorados e registrados os resultados obtidos.

Um conjunto de atividades de testes estará sendo realizado no decorrer de cada uma destas fases. Todavia, considerando os custos envolvidos em uma abordagem que postergue o início destas atividades para somente após o final da construção do software, se torna bastante relevante antecipar a execução destas atividades. Isto porque é possível realizar atividades de verificação e validação antes do projeto apresentar código pronto. O planejamento das atividades de testes e a inspeção de documentos de requisitos do projeto, *design* de programas e código podem produzir resultados significativos em termos de custo/benefício.

O V-MODEL fornece um meio de integração entre as atividades de desenvolvimento e as atividades de teste. Este modelo é apresentado na figura 2-1.

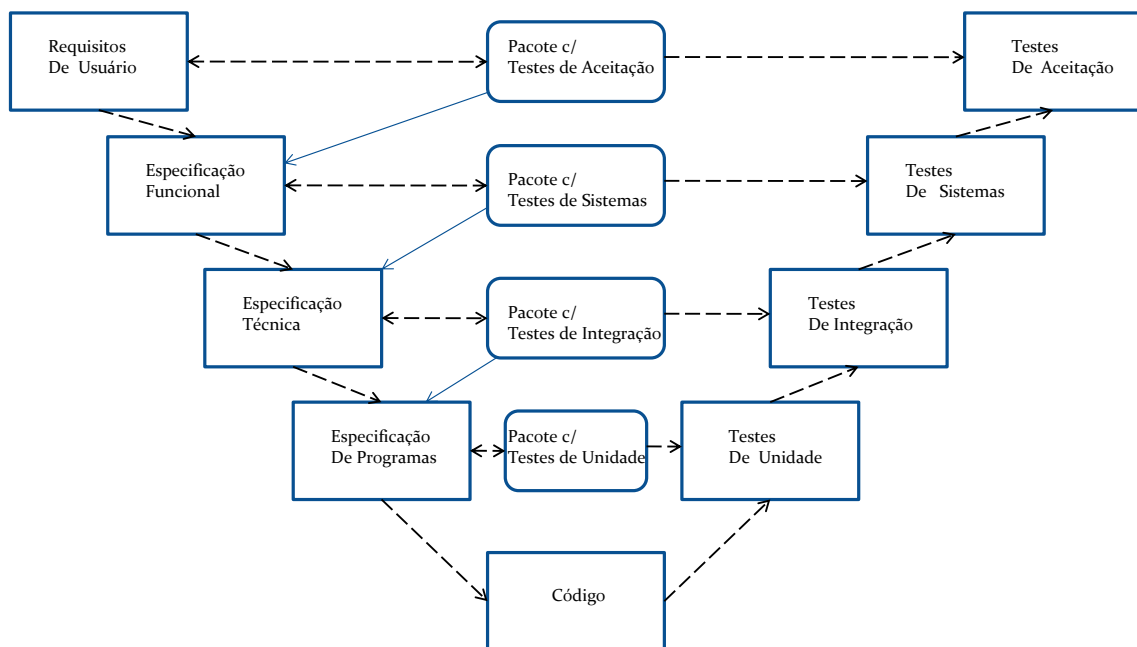


Figura 2-1 Modelo V (adaptado de DAVIS (2000))

Nas etapas de elicitação e análise dos requisitos de usuário e de especificação funcional, os testadores estarão acessando os documentos de requisitos, e interagindo

com o usuário para desenvolver uma versão inicial dos testes de sistemas e aceitação, baseados nos requisitos funcionais, requisitos de qualidade, e na especificação do comportamento do sistema. O detalhamento dos requisitos permitirá verificar se estes são testáveis.

Em cooperação com o gerente, os analistas de testes poderão iniciar o planejamento do projeto de teste, quando será definida a estratégia de testes (os níveis de testes que serão empregados) e os objetivos do teste. Os entregáveis desta fase são: uma versão preliminar do plano do projeto de testes (com suas estimativas de custo, riscos, prazos) e os planos de testes de aceitação e testes de sistemas contendo uma versão preliminar dos casos de testes.

Na etapa de especificação técnica a arquitetura do sistema está sendo descrita e a estrutura dos componentes já é conhecida – tanto em sistemas baseados no paradigma de orientação a objetos como em sistemas baseados em procedimentos. Os analistas de testes podem interagir com os projetistas para verificar a Testabilidade do projeto e realizar o planejamento dos testes de integração. Caso o projeto esteja detalhado em baixo nível, já é possível iniciar o projeto dos testes de unidade. Os entregáveis nesta fase são: o plano do projeto de testes revisado, o plano de testes de aceitação e sistema revisados e o plano de testes de integração.

Na etapa de especificação dos programas todos os planos são passíveis de uma revisão, a partir do conhecimento mais detalhado que os analistas de testes e testadores possuem e, à medida que o software vai ficando disponível para execução. Nesta etapa, a atividade de especificação se concentra mais nos testes de unidade, e segue concorrentemente à atividade de codificação. Os entregáveis são: todos os planos anteriores revisados, acrescidos do plano de testes de unidade, e dos marcos das atividades de testes que serão realizadas na sequência.

Após a finalização da etapa de codificação a equipe está preparada para executar os testes de unidade, integração, sistemas e aceitação em uma estratégia *bottom-up*. Nesta etapa os entregáveis são: o código testado, os logs de testes e os relatórios de incidentes de teste, as métricas dos testes, o resumo das atividades de testes (BURSTEIN, 2003).

A adoção do modelo V conjuntamente com revisões pode trazer mais efetividade ao processo de detecção de defeitos. Os tipos de revisões propostas foram derivados de um trabalho do British Computer Society Specialist Interest Group *in Software Testing* (DAVIS, 2000):

Revisão do tipo 1 - Visa assegurar que o produto (documento, programa, design da tela, procedimento) está sendo construído com os padrões adequados, e contém as características que se espera deste tipo de produto. É um teste estático

para garantir que o produto está internamente consistente, preciso e completo. Se por exemplo, o produto é uma especificação funcional devemos garantir que ele possua tudo que uma especificação deste tipo deve conter – se está livre de erros, incoerências, ambiguidades e omissões.

Revisão do tipo 2 - É uma atividade para verificar se o produto que está sendo gerado na fase atual está coerente com as especificações do(s) artefato(s) produzido(s) pela atividade anterior. Obriga o revisor examinar o(s) produto(s) da etapa anterior e avaliar se o que está sendo gerado na fase atual está completo e consistente com o que tinha sido feito até então.

Revisão do tipo 3 – Visa verificar se existe informação suficiente no produto para que a próxima atividade possa ser executada com sucesso. Depois de uma revisão deste tipo não pode existir questões pendentes.

2.3.2 Processo de Testes (HASS, 2004)

O modelo V não prevê como lidar com testes adicionais que se façam necessários por mudança de critério de cobertura de alguma funcionalidade, o que é previsto no processo de testes formulado por Hass (HASS, 2004).

Segundo Hass, um processo é um conceito importante no âmbito das atividades de desenvolvimento de software. Quando as atividades são organizadas em descrições de processo, estas podem ser comunicadas, seguidas, observadas, e melhoradas. A base para a compreensão do que é o teste, é, portanto, a compreensão do processo de testes.

O modelo proposto é baseado nas seguintes premissas:

- ✓ As atividades de testes nem sempre deverão ser executadas em uma única ordem sequencial. As atividades de monitoramento, controle e replanejamento de testes serão executadas durante a execução do projeto de teste.
- ✓ O processo genérico de testes proposto é iterativo. As atividades são executadas até que o critério de saída (escopo de cobertura) seja atingido.

Segundo Hass, um processo de testes é constituído das atividades de planejamento inicial, monitoramento controle, replanejamento, projeto e desenvolvimento dos testes, execução, avaliação e reporte dos resultados obtidos e, encerramento das atividades de testes. Na figura 2-2 estão representadas as atividades que compõem o processo proposto.

As duas primeiras iterações são realizadas após a execução das atividades de planejamento inicial dos testes e projeto e de projeto e desenvolvimento, ambas após a execução dos testes:

1. Quando o defeito que originou a falha é corrigido no artefato do software, deverá ser reexecutado o mesmo procedimento que revelou a falha, e opcionalmente a execução de uma suíte de testes de regressão.
2. Quando é revelada uma inconsistência no Procedimento de Teste. Neste caso o procedimento deverá ser corrigido e atualizado, sendo reexecutado na sequência.

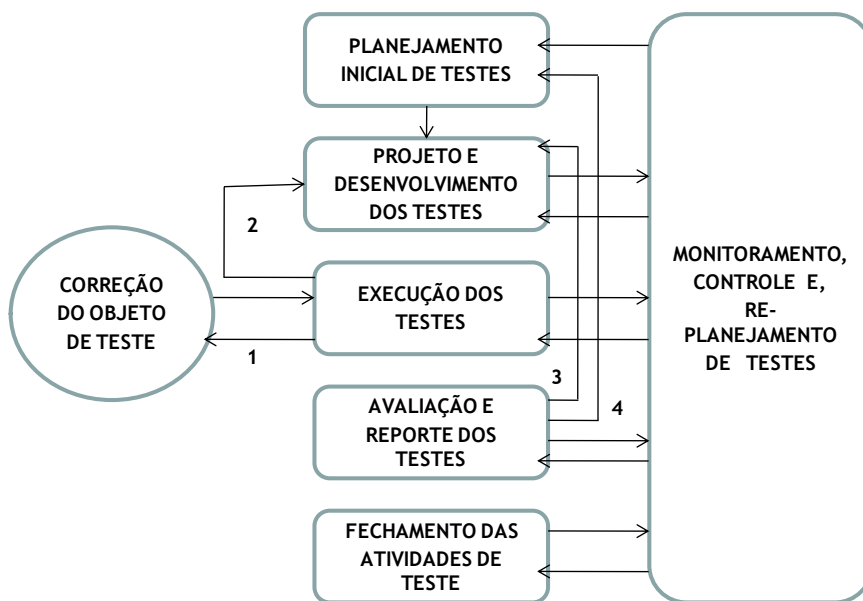


Figura 2-2 Processo Genérico Iterativo Proposto por HASS.

Um ciclo iterativo poderá ocorrer novamente após a execução da atividade de avaliação e relato de testes:

3. Quando é avaliada a necessidade de aumento de cobertura dos testes, deverão ser especificados novos Casos de Testes, a partir da execução da atividade de Projeto e de Desenvolvimento de Testes.
4. Ao avaliar os resultados obtidos, o Gerente de Testes pode considerar a necessidade de modificar os critérios de saída no Plano de Testes (seja diminuindo o escopo da cobertura, seja aumentando).

Hass realça a necessidade do gestor do processo de testes ter consciência que os artefatos de testes são apenas meios para se alcançar os objetivos dos testes, e são resultados da execução das atividades, escolhas de alternativas de *design*, e

outras tomadas de decisão no decorrer do processo. E lista o conjunto de artefatos a serem gerados e utilizados no processo:

- ✓ Plano de Testes do Sistema.
- ✓ Especificação do Teste.
- ✓ Especificação dos ambientes a serem utilizados nos Testes.
- ✓ Os dados a serem utilizados nos testes e o ambiente real.
- ✓ Histórico dos Testes (Log).
- ✓ Relatório de Progressos dos Testes
- ✓ Relatório de Resumo dos Testes
- ✓ Relatório sobre a experiência dos Testes (lições aprendidas)

Todavia, o exame da relação destes artefatos e seu conteúdo pode nos auxiliar a compreender melhor os objetivos das atividades e quais são as transformações associadas a cada tarefa.

2.3.3 Processos Verificação e Validação (MPS.BR)

O processo de verificação proposto pelo modelo MPS.BR tem como objetivo confirmar que cada serviço ou produto de trabalho do projeto ou do processo atende de forma apropriada os requisitos especificados. Sua avaliação é realizada a partir de um conjunto de resultados esperados e de atributos de processo.

Cada resultado esperado é um resultado observável do sucesso do alcance do propósito de processo (ISO/IEC, 2008a). Um resultado pode ser: um artefato produzido, uma mudança significativa de estado e o atendimento das especificações, como por exemplo: requisitos, metas etc.

Um atributo do processo é uma característica mensurável da capacidade do processo aplicável a qualquer processo (ISO/IEC, 2003). E a capacidade do processo reflete o quanto o processo está institucionalizado na organização e o grau de refinamento em que está sendo executado.

O processo de verificação interage com os processos de Projeto e Construção de Produto (PCP), Desenvolvimento de Requisitos (DRE) e Integração do Produto (ITP). A intersecção com o processo DRE aparece no resultado esperado utilizado para avaliar se a análise dos requisitos foi desenvolvida de forma a assegurar sua verificação. Os produtos gerados durante o ciclo de vida do desenvolvimento devem ser avaliados para que seja verificado se a implementação está de acordo com o que foi especificado na etapa de projeto (design). A iteração com o processo ITP acontece durante as várias integrações de

componentes e rotinas realizadas no decorrer do projeto quando a equipe deverá realizar os testes de integração.

As avaliações de qualidade devem ser realizadas ao longo do ciclo de vida do software e iniciam com a verificação se os requisitos estabelecidos podem ser atendidos, evoluem através da avaliação dos produtos intermediários até a realização da versão final do produto, com o intuito de garantir que o produto está sendo construído de acordo com o especificado.

No guia de implementação são descritos dois métodos de verificação: revisão por pares e testes. Revisão por pares é um teste estático através do qual o artefato selecionado para verificação é avaliado por um dos participantes do projeto. Com a restrição que o revisor escolhido não poderá ser o mesmo membro da equipe responsável pela criação do artefato (LAITENBERGER et al., 2002). Testes podem ser descritos como um processo utilizado para revelar defeitos em software, e estabelecer que o mesmo atingiu um determinado grau de qualidade.

Os resultados esperados do processo de Verificação visam avaliar se: (a) existe uma estratégia para identificar e selecionar quais artefatos deverão ser verificados. (b) cronogramas e métodos são estabelecidos e profissionais são alocados compondo uma estratégia para implementação de um processo de verificação. Os métodos de verificação podem ser estáticos (revisões) ou dinâmicos (testes). (c) um ambiente é estabelecido com a disposição de ferramentas, recursos de hardware, infraestrutura de rede, e um conjunto de procedimentos são definidos para verificação dos produtos de trabalho (d) as atividades de testes, incluindo testes e revisão por pares são executados (e) os defeitos são identificados e registrados (f) os resultados obtidos a partir da execução das atividades de verificação são analisados e compartilhados para os *stakeholders*.

O propósito do Processo de Validação é confirmar que o software atenderá ao uso pretendido ao ser instalado no ambiente para o qual foi desenvolvido. Sendo assim, o objetivo deste processo é assegurar que o produto correto está sendo desenvolvido.

Assim como acontece com o processo de Verificação, os resultados esperados do processo de Validação estão relacionados aos processos Desenvolvimento de Requisitos (DRE) e Integração de Produto (ITP).

As técnicas comumente utilizadas em atividades de validação são: Testes de aceitação, protótipos, simulações, *model checking* e utilização de cenários de caso de Uso.

Os alcances dos resultados esperados do processo de Validação são bastante semelhantes àqueles relacionados ao processo de Verificação: (a) os produtos a validar deverão ser identificados; (b) deverá ser desenvolvida uma estratégia contemplando cronograma, participantes envolvidos e métodos para validação; (c) um ambiente para validação e um conjunto de critérios e procedimentos deverão ser estabelecidos; (d) as atividades previstas no processo de validação são executadas para assegurar que o produto esteja pronto para uso no ambiente operacional pretendido (e) problemas são identificados e registrados (f) os resultados obtidos através da execução das atividades de validação são avaliados e comunicados aos envolvidos (g) foram geradas evidências que os produtos de software desenvolvidos estão prontos para o uso pretendido.

2.3.4 Processo de Testes (Dias Neto, 2006)

No artigo de Hass (2004) não estão detalhados os conteúdos de cada um dos artefatos do processo. Porém, a partir da descrição de cada um deles, verificamos correspondência com os artefatos descritos em outros processos de teste, como aqueles sugeridos por Dias Neto e Travassos (2006), cujo processo está baseado na norma IEEE-829/1998.

Segundo esta versão da norma, os artefatos a serem considerados pelo gerente do processo de testes são: (a) Plano de Testes do Sistema: contém o propósito dos testes, o escopo a ser testado (módulos, funcionalidades), as referências a documentos e artefatos de desenvolvimento (como por exemplo, Casos de Uso, Cenários, Requisitos), os itens sobre teste, os níveis de testes (aceitação, sistema, integração, unidade), as características a serem testadas e aquelas que não serão testadas, as necessidades ambientais, as necessidades de treinamento, o cronograma das atividades de teste, os riscos e as contingências (b) Especificação de *Design* dos Testes: itens a serem testados, abordagem dos testes (por exemplo: serão gerados grafos para todos os fluxos principais e alternativos dos casos de uso, e para cada um destes grafos um conjunto de procedimentos e casos de testes será desenvolvido), relação dos casos de testes e procedimentos, critério de aprovação do projeto de testes (por exemplo, todas as *features* serão consideradas ok se apenas todos os casos de testes passarem com sucesso) (c) Especificação dos Casos de Testes – Contém o nome do Item sob testes que estará sendo exercitado através do

caso de teste, entradas e saídas esperadas, ambiente necessário para a realização do teste, dependência com outros Casos de Testes (d) Especificação dos Procedimentos de Testes – propósito do procedimento, Requisitos para execução do procedimento (ambiente), passos do procedimento (e) Histórico dos Testes (Log) – Lista dos eventos que ocorreram durante a execução das atividades de teste: criação de Casos de Teste, modificações no hardware e software de teste, anomalias/eventos inesperados, registro de sucesso ou insucesso na execução de Casos de Testes (f) Relatórios de Incidentes nos Testes – Uma falha sistêmica pode gerar um incidente nos testes que neste caso pode não ser decorrente de existência de um defeito na aplicação que está sendo testada. Relatórios de incidentes servem para que sejam avaliadas estas ocorrências. Ele contem um detalhamento da execução dos testes com dados de entrada, resultados esperados, resultados obtidos, eventuais anomalias, procedimento executado com ultimo passo realizado, ambiente de execução, testador, tentativas de re-execução (g) Relatório de Resumo dos Testes - relação dos artefatos desenvolvidos (plano de teste, design de teste, casos de teste, etc.), lista dos incidentes de testes (apenas data e identificador), registro se os critérios de cobertura planejados foram satisfeitos, e em caso de relaxamento do critério explicar a razão, cronograma atualizado com o realizado em cada uma das atividades, registro dos incidentes que não foram resolvidos.

Estes artefatos também deverão conter um identificador único, data e hora, e membro da equipe responsável pela geração do artefato.

Neste processo, a abordagem de apoio para planejamento e controle dos testes foi baseada em um princípio de sistematização e na definição de uma documentação (artefatos IEEE-829/1998). A documentação teve com objetivo fornecer evidências sobre a realização de determinada atividade e, ao mesmo tempo, facilitar a comunicação entre os envolvidos.

A sistematização tem como objetivo principal organizar o conjunto de atividades e tarefas a serem realizadas durante os testes. E com este objetivo foi modelado um processo de testes cujas macro atividades foram originadas de um outro processo apresentado por PFLEEGER (2004) e, constituído das seguintes atividades: Planejamento dos Testes, Projeto dos Testes, Especificação dos Casos de Teste, Definição dos Procedimentos de Teste, Execução dos Testes e Análise dos Resultados dos Testes. Também foram modelados três perfis a serem associados às atividades: Gerente de Teste, Projetista de Testes e Testador. (DIAS NETO e TRAVASSOS, 2006).

2.3.5 Norma IEEE-829/2008

A revisão da norma IEEE-829 publicada em 2008 descreve a compatibilidade com o padrão IEEE/EIA Std 12207.0-1996 de forma mais detalhada, e para cada um dos processos exibidos na figura 2-3 enumera cada uma das atividades de gerencia, aquisição, fornecimento, desenvolvimento, operação e manutenção relacionadas aos esforços de teste.

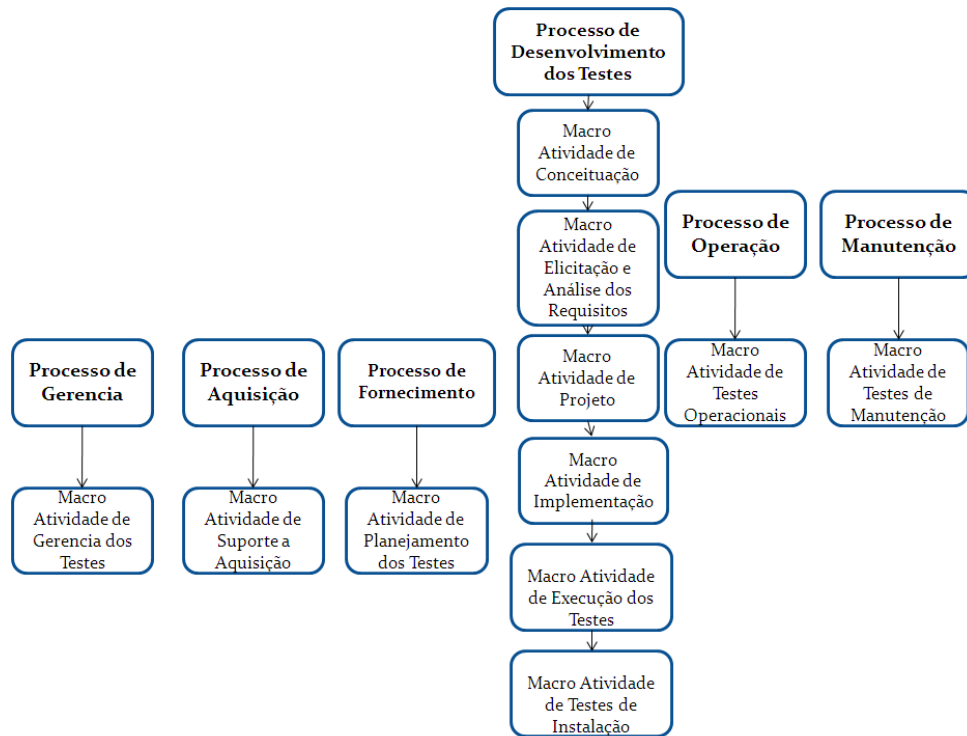


Figura 2-3 Compatibilidade norma IEEE-829/2008 c/ IEEE/EIA Std 12207.0-1996.

Apesar da versão IEEE-829/2008 apresentar um maior nível de detalhamento, ao compararmos esta versão da norma com a versão publicada em 1998 utilizada por Dias Neto (2006) constatamos que os artefatos permanecem os mesmos e as atividades relacionadas a design, especificação e execução de testes preservam a finalidade e o mesmo sequenciamento.

A versão IEEE-829/2008 oferece dois tipos de utilização: (a) uso parcial (b) uso completo.

Ao optar pelo uso completo ao iniciar um determinado projeto, o responsável pelos testes deverá estabelecer o nível de integridade mais adequado aos riscos e restrições de segurança deste projeto e utilizar uma tabela da especificação para que possa obter a lista dos artefatos adequados ao nível de integridade selecionado. Pois, cada nível de integridade possui uma relação de atividades requeridas para inclusão na adaptação do processo, que por sua vez estão associadas a um conjunto de artefatos a serem utilizados.

Caso opte pelo uso parcial do padrão quatro atividades deverão necessariamente ser executadas. Conforme exibido na figura 2-4.

No escopo deste trabalho estarão sendo utilizadas as atividades do uso parcial do padrão pois estas suportam a realização do planejamento do projeto de teste, controle e monitoramento e as atividades de design e especificação de procedimentos e Casos de Testes, contemplando todas as atividades mapeadas às práticas ágeis (Apêndice D).

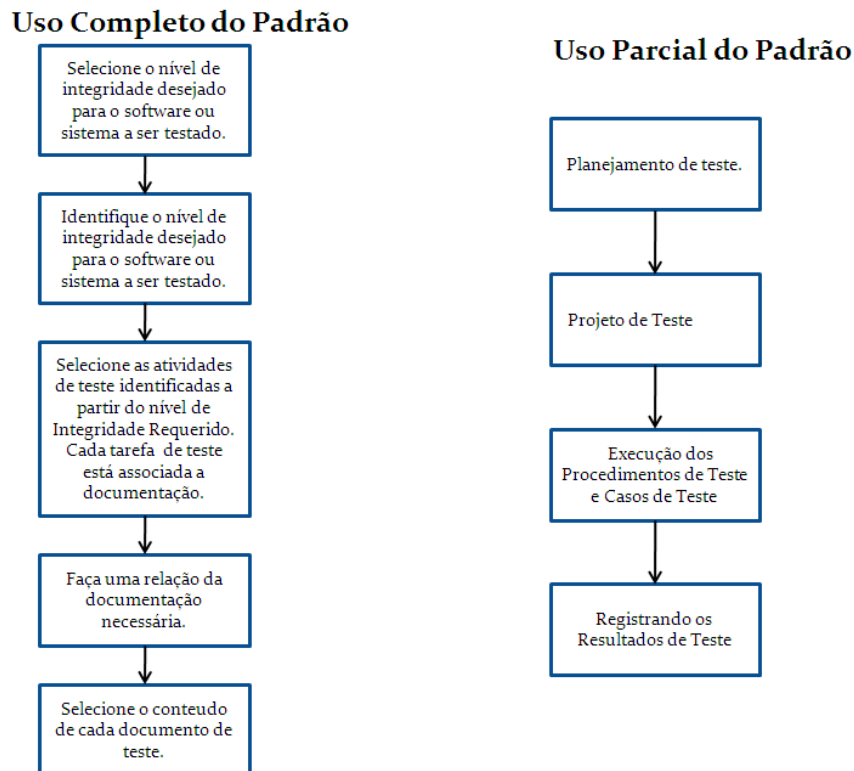


Figura 2-4 Estratégias de uso do padrão fornecidas pela norma IEEE-829/2008

2.3.6 Críticas ao Modelo V

O modelo V foi estendido e adaptado gerando algumas versões. Uma de suas derivações (*W-Model*) enfatiza a necessidade de execução dos testes estáticos e não obriga que todo plano de testes deva ser o resultado apenas da análise dos documentos gerados pelas atividades de elicitação e análise de requisitos, permitindo a utilização de fontes adicionais. (HERZLICH, 1993).

Rex Black faz uma crítica à ênfase que por vezes era concedida às etapas de análise e projeto conjuntamente com as atividades correlatas de planejamento de testes a serem realizadas, fazendo com que o projeto gastasse mais tempo nestas atividades e encurtando os recursos disponíveis para a realização dos testes. (BLACK, 2002)

Marik questiona fortemente a utilização do modelo V, enfatizando que ele divide as atividades de testes em fases com limites rígidos, condicionando a execução de testes de uma determinada fase somente após a execução da fase anterior. Critica também em alguns casos o uso de algumas técnicas de integração por conta do esforço de implementação necessário, como é o caso de *stubs* e *drivers* para a emulação de unidades e rotinas que ainda não foram desenvolvidas. E que por vezes, pode ser mais vantajoso considerar a opção de não implementar estas técnicas, e assumir os riscos de postergar os testes para um momento mais à frente, quando todas as rotinas estão disponíveis para testes. E que o modelo deveria facilitar a atualização do *design* de testes quando novas informações surgissem (MARIK, 2000).

GoldSmith e Graham contrapõem Marik, afirmando que tão logo estejam disponíveis informações sobre os requisitos e funcionalidades, já devem ser identificadas as condições de testes e casos de testes do projeto, e isto pode ser aplicado a qualquer ou todos os níveis de testes propostos no Modelo V. Sendo fundamental não interpreta-lo de forma rígida. O gestor deve usa-lo como um *guideline* sobre a necessidade de definir o que deve ser feito (requisitos), descrever como fazer (*design*), antes de passar para a etapa de execução (código). Cada nível de desenvolvimento tem um nível de testes correspondente, e os testes podem ser concebidos no início da etapa de desenvolvimento de todos os níveis.

A modularização das funcionalidades, e separação em ciclos curtos, no desenvolvimento iterativo permite a execução de vários ciclos curtos das atividades do modelo (execução de pequenas iterações na forma de vários modelos Vs). Contudo, independentemente do tamanho do ciclo, a sequência das atividades *what* (“o-que-deve-ser-feito”), *how* (“como”) e, *do* (“execução”) é essencial para obtenção de qualidade com um custo mínimo e um prazo exequível. Também é importante garantir que os objetivos de cada nível foram de fato foram cumpridos (GOLDSMITH e GRAHAM, 2002).

De todo modo, Marik não chega a propor um processo formal de teste. Contudo registra algumas preocupações que devem ser consideradas na condução e definição de um processo de teste:

- ✓ A cada entrega de código deve ser analisada a possibilidade de execução de testes desta porção de código. O resultado desta análise de custo benefício pode decidir por postergar os testes para um momento posterior. Contudo tanto a decisão como a entrega deve ser registrada.

- ✓ Outras fontes de informação, além dos documentos de requisitos devem ser consideradas. A comunicação com desenvolvedores, projetistas, usuários deve ser incentivada.
- ✓ As execuções de testes devem ser consideradas como fontes de informação do sistema. Um bom testador aprende e intui eventuais fraquezas do sistema a partir dos resultados de uma rodada (*trial*) de um conjunto de casos de teste.
- ✓ A alta frequência de replanejamento deve ser considerada, dado a quantidade de ajustes, mudanças que chegam do processo de desenvolvimento.

Na evolução desta crítica, Marik formulou uma abordagem que visa categorizar as atividades de testes a partir de 2 dimensões, cada uma contemplando duas perspectivas. A primeira dimensão é composta de duas categorias: Crítica ao Produto e Suporte ao Time. Nela são colocados em lados opostos, os testes que exercitam o sistema, para verificar a existência de defeitos na implementação dos requisitos funcionais e não-funcionais, e os testes que tem a intenção de verificar aspectos de *design*, implementação e programação e desta forma dariam suporte ao time.

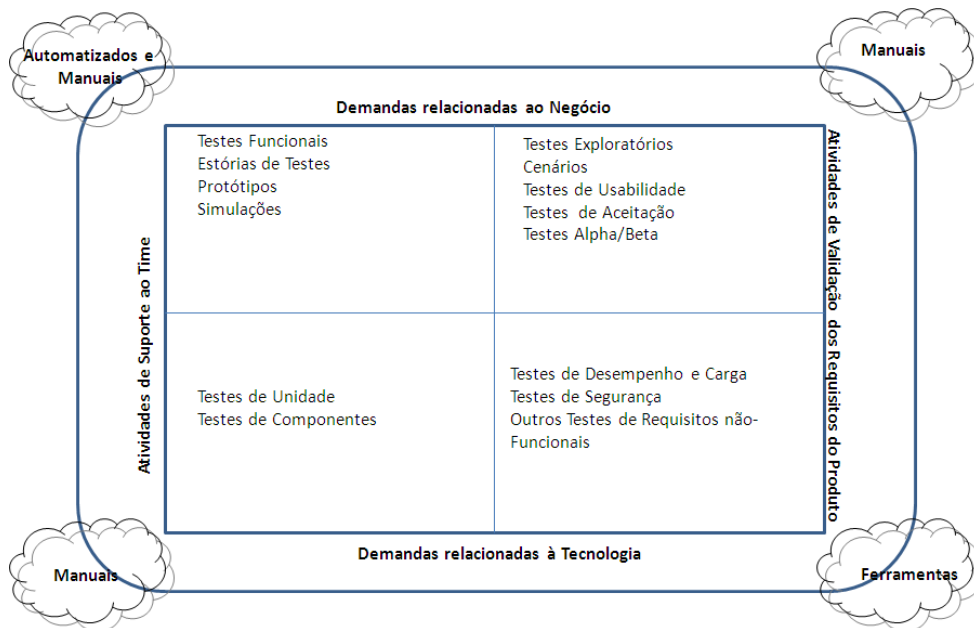


Figura 2-5 Agile Testing Quadrants - Marik.(CRISPIN & GREGORY, 2009)

A segunda dimensão é composta também de duas categorias: Aspectos de Negócio e Aspectos de Tecnologia. Da mesma forma nesta dimensão são classificados em categorias distintas os testes que estão associados à aspectos de tecnologia (desempenho, carga, unitários e integração) dos testes associados ao

negócio (funcionais, aceitação, usabilidade e outros). Na figura 2-5 é mostrado o *Agile Testing Quadrants*.

2.3.7 Processo de testes selecionado.

Abrantes (2012), ao comparar as atividades de teste propostas por Hass (2004), Davis (2000) e aquelas baseadas na norma IEEE-829/1998 especificadas por Dias Neto (2006) registrou a ausência da atividade de Monitoramento, Controle e Replanejamento e da atividade de Encerramento dos Testes pelo processo baseado na norma IEEE-829. Observou também que apesar do processo modelado por Dias Neto (2006) não especificar formalmente estas atividades, é prescrito o uso de documentação específica para apoiar avaliações entre o escopo planejado e realizado, bem como a análise das ocorrências durante a execução dos testes, incluindo uma atividade de coleta e registro dos dados de testes.

Quanto aos produtos de trabalho propostos pelos processos estudados, ao analisar os artefatos do processo genérico proposto por DAVIS (2000) foi observado aderência em relação às atividades e produtos contidos nos outros dois processos, sendo registradas diferenças apenas nos níveis de detalhe dos artefatos produzidos.

Dessa forma, desde que não foi observada nenhuma mudança significativa entre a versão IEEE-829/2008 com a versão anterior IEEE-829/1998, utilizada na especificação das atividades do processo de Dias Neto (2006), estaremos utilizando-o no âmbito deste trabalho, incluindo apenas uma atividade de monitoramento, controle do projeto de teste e outra de encerramento das atividades.

A aplicação deste processo em organizações cujos processos foram avaliados a partir de um dos modelos de maturidade como MPS.BR e CMMI deverá ser acompanhada pelo grupo de processos, de forma a preservar a aderência aos resultados esperados dos processos implementados após às mudanças sugeridas.

Com a definição do modelo de processo a ser utilizado, a próxima etapa da pesquisa visou elencar qual linguagem de modelagem de processo poderia ser usada com o propósito de fornecer diferentes níveis de apoio ao pessoal envolvido, e refletir rapidamente mudanças no processo, de acordo com a prática “Visibilidade de Projeto” (prática descrita em Apêndice D).

2.3.8 Definição da linguagem para documentação do Processo

Na figura 2-6 estão dispostas as atividades análise, projeto, instanciação, simulação, execução e avaliação que compõem o ciclo de vida de um processo de

software, os perfis envolvidos e os recursos de suporte à execução destas atividades, baseado em um esquema PDCA².

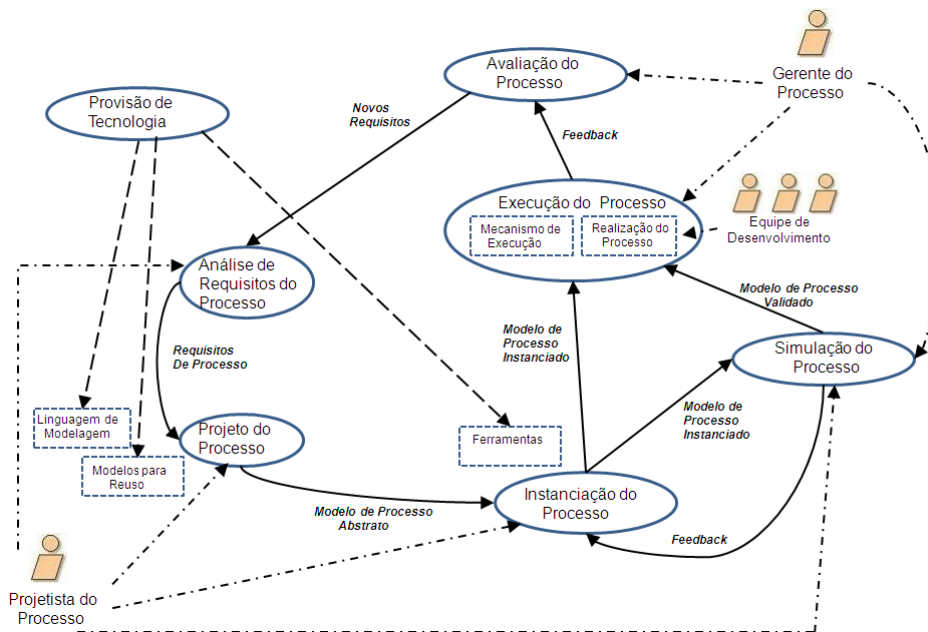


Figura 2-6 Meta-processo para criação e utilização de um PDS. Adaptado de Reis (2003)

Desde que processos de software por vezes apresentam estruturas complexas, foram desenvolvidos pelos pesquisadores um número de linguagens e formalismos que facilitam a representação das características presentes em um processo desta natureza (FUGGETTA, 2000):

- ✓ Atividades que deverão ser realizadas para que sejam alcançados os objetivos do processo;
- ✓ Perfis de cada *stakeholder* envolvido no processo;
- ✓ Estrutura e características do artefato a ser utilizado ou gerado por determinada atividade do processo, e;
- ✓ Ferramentas utilizadas.

Este conjunto de linguagens e formalismos chamado de linguagens de modelagem de processos (*Process Modeling Languages – PML*) tem como objetivo suportar as etapas do ciclo de vida de um processo de software, provendo elementos para decomposição das várias atividades executadas pelas organizações, em seus processos de desenvolvimento de software (PDS).

As linguagens para modelagem de processos podem ser utilizadas para diferentes fins (FUGGETTA, 2000):

² Método iterativo de gestão (P-Plan, D-Do, C-Check e A-Act) utilizado para controle e melhoria contínua de processos.

- I. Compreensão do processo - A PML pode ser usada para representar de um modo preciso como um processo está estruturado e organizado. Isto pode ser útil para eliminar inconsistências na especificação do processo.
- II. Projeto do processo – De forma proativa, uma PML pode ser usada para conceber um novo processo, ao descrever a sua estrutura e organização.
- III. Formação e Treinamento - Uma descrição precisa do processo pode ser útil para ensinar os procedimentos da empresa e operações para o pessoal recém-contratado.
- IV. Simulação de processo e otimização – A descrição do processo pode ser simulada para avaliar possíveis problemas, gargalos e oportunidades para melhoria.
- V. Processo de apoio - Uma descrição precisa do processo de pode ser interpretada e usada para fornecer diferentes níveis de apoio para as pessoas que atuam no processo.

Considerando estes recursos e as necessidades de entendimento do processo de testes corrente na organização a ser estudada, e de formulação de um projeto contendo as mudanças propostas para o processo de testes original da organização, as vantagens listadas nos itens I, II, III e V acima sugerem que a adoção de uma PML poderá ser de grande valia para esta fase do trabalho.

Considerando o provisionamento de recursos tecnológicos foram avaliadas duas especificações propostas pela OMG para modelagem dos processos de testes a serem utilizados nesta pesquisa.

O SPEM 2.0 (*Software & Systems Process Engineering MetaModel*) é um metamodelo para a especificação de processos de software, concebido para ser utilizado na descrição de um processo de desenvolvimento de software ou uma família relacionada de processos de desenvolvimento de software.

E o BPM (*Business Process Management*) cuja abordagem para o problema da modelagem de processos de desenvolvimento de software consiste em tratá-lo como um processo de negócio onde o negócio é desenvolver software - conforme exposto em (ARAÚJO, 2004). Uma das principais ferramentas nesta prática é a BPMN (*Business Process Model and Notation*).

Segundo a sua especificação (OMG, 2010), a versão 2.0 da BPMN tem como objetivo principal fornecer uma notação que seja facilmente compreensível por todos os usuários de negócio, desde o analista que cria as versões iniciais do processo, passando pelos desenvolvedores, responsáveis por implementar a tecnologia que executará estes processos até, finalmente, às pessoas de negócio que vão manter e monitorar estes processos.

Em uma primeira avaliação percebe-se que o BPMN é uma notação cujo ponto forte é a representação dos fluxos de atividades, com todos os recursos de projeto de fluxos alternativos e de exceção. Seu principal foco é a representação de Processos de Negócio que são desenhados para serem seguidos por cada instância do processo, podendo sua execução ser acompanhada através do WSBPEL.

Já o SPEM define um meta-modelo através do qual é possível modelar qualquer PDS, fornecendo um conjunto de estereótipos que facilitam a documentação e o *design* de um processo de desenvolvimento. Os recursos de *Capability Pattern* e *Process Delivery* propiciam o reuso e *tailoring* de processo, apoiando o desenvolvimento de processos especializados. A criação e manutenção de um processo através da implementação de melhorias pode ser facilitada pela adoção do SPEM (Figura 2-6).

Em uma análise preliminar dos recursos existentes no SPEM pode-se elencar algumas vantagens em sua utilização:

- Padrões de Processos - Permite a criação de catálogos de processos e disposição de padrões de processos, podendo ser adaptados para necessidades específicas.
- *Process Building Blocks* - Os elementos de método são organizados em pedaços reutilizáveis de processo denominados *capability-patterns*, promovendo uma abordagem consistente de desenvolvimento para necessidades comuns dos projetos.
- Recursos Específicos para ES - Recursos específicos do domínio em Desenvolvimento de Software. Oferece alguns estereótipos de templates usados em engenharia de software.
- Base de Processos - Facilita a criação e disposição de uma base de conhecimento dos processos (*Tailoring*)

Uma carência percebida na especificação SPEM diz respeito à falta de apoio à execução de um processo de software. No documento publicado pela OMG são sugeridas duas formas de acompanhamento da execução: a adoção de sistemas de planejamento e acompanhamento de projeto e, o uso de modelos externos de comportamento para workflow - seja através do uso de diagrama de atividades UML 2.0 ou com BPMN/BPDM.

O projeto *Open-Source Eclipse Process Framework* (EPF) mantido pelo *Eclipse Foundation* provê uma ferramenta para modelagem de processos constituída de um conjunto de funcionalidades que facilitam a modelagem dos processos de desenvolvimento, a partir de uma evolução do padrão SPEM 1.1, sendo um dos objetivos do projeto EPF obter compatibilidade plena à versão 2.014 SPEM.

Em um relato de experiência no uso da ferramenta *Eclipse Process Framework Composer* (EPF) foram apontadas algumas vantagens em sua adoção em uma Organização CMMI nível 3 (FURTADO *et al*, 2008):

- ✓ Facilidade do acesso às informações do processo pelas equipes do projeto e do cliente, melhorando a visibilidade das adaptações;
- ✓ Reutilização dos ativos do processo: a biblioteca de ativos de cada subprocesso fica disponível para todos os membros do SEPG (Software Engineering Process Group);
- ✓ Maior consistência das informações dos processos adaptados para os projetos em relação às atualizações do processo organizacional, pois as mesmas são geradas de forma automática;
- ✓ Redução do retrabalho dos processos adaptados para os projetos e maior agilidade na disponibilização das melhorias do processo organizacional;

Neste mesmo artigo são registrados alguns fatores críticos para implementação:

- ✓ Necessidade de definir responsáveis na Organização para prover suporte no uso da ferramenta e utilizar uma ferramenta colaborativa para compartilhar experiências do uso.
- ✓ Necessidade de estruturar o processo organizacional de forma a aperfeiçoar o reuso de elementos do processo de desenvolvimento.
- ✓ Curva de aprendizado na ferramenta e estruturação dos elementos do processo organizacional;

Considerando este relato de uso, a capacidade de documentação e visualização dos detalhes do processo de desenvolvimento de software foi escolhido o uso do SPEM e o uso da ferramenta EPF para documentar os processos de testes e os demais processos de software envolvidos no escopo deste trabalho.

2.4 Conclusão

Neste capítulo foram registrados alguns desafios a respeito da conceituação do que vem a ser agilidade, e sobre os modelos de processos de testes propostos no âmbito da pesquisa em Engenharia de Software.

Na revisão da literatura sobre processos de teste foram coletados esforços no sentido de prover mais agilidade às atividades prescritas, seja através de um conjunto de diretrizes para adaptação do processo de testes de acordo com o nível de criticidade do sistema, ou pela proposição da iniciação das atividades de planejamento

de testes, tão logo já existam informações à cerca dos requisitos e funcionalidades, ou através da realização de revisões a serem realizadas nos documentos gerados nas etapas de elicitação, análise de requisitos e projeto, ou ainda pela execução de ciclos curtos de iteração do modelo V.

Também foi selecionado um modelo de processo de teste a ser utilizado no âmbito desta pesquisa, e elencado uma linguagem de modelagem capaz de representar os artefatos, atividades, perfis e tarefas contidas no processo a ser utilizado como padrão.

Na revisão da literatura um conjunto de práticas ágeis foi recuperado com o intuito de buscar por novas ocorrências além daquelas já propostas por Abrantes (2012), sem que tenha sido registrado o relato de uma prática que não estivesse associada ao conjunto proposto originariamente.

Como desdobramento destas buscas, o desenvolvimento de um protocolo de *quasi*-revisão sistemática, com o objetivo de buscar por relatos de adoção destas características de agilidade, especificamente em atividades de teste, poderá contribuir para agregar novas evidências e novas perspectivas para este campo da pesquisa.

No próximo capítulo estará sendo descrito o protocolo de uma revisão e sua execução.

3 Características de Agilidade e Processos de Testes – *Quasi*-Revisão sistemática da literatura

Neste capítulo é apresentado o planejamento e o resultado da execução de uma quasi-revisão sistemática sobre características de agilidade.

3.1 Introdução

Revisão sistemática é um instrumento fundamental para promover e habilitar práticas baseadas em evidências, pois reúnem e combinam os resultados de vários estudos.

Uma pesquisa conduzida através de um processo sistemático de revisão segue uma sequencia de etapas de um método através de um protocolo definido a priori. Este instrumento é construído a partir de uma questão central, a qual representa o núcleo da investigação e é expresso usando conceitos e termos que devem ser endereçados à questão estruturada. As etapas da metodologia, as estratégias para recuperação de uma evidência, o foco da questão são definidas de forma explicita. De forma que outros profissionais possam reproduzir o mesmo protocolo e também sejam capazes de julgar a adequação dos padrões escolhidos (Dyba et al, 2007).

Os resultados obtidos na revisão inicial da literatura motivaram o estabelecimento de um protocolo de revisão sistemática, cujo objetivo é buscar por relatos de adoção de características de agilidade, especificamente em atividades de teste, com o intuito de contribuir para agregar novas evidências nesta área de pesquisa.

Um dos objetivos desta revisão é averiguar a existência de outras práticas e características que porventura não tenham sido detectadas nas revisões realizadas por Abrantes (2012). E visa apoiar e confirmar a pertinência do encaminhamento de um estudo de campo sobre a adoção de práticas ágeis em um processo de testes formalizado, em uma determinada organização selecionada para este fim.

3.1.1 Protocolo de revisão sistemática

Um dos critérios utilizados para a definição do protocolo da revisão foi realizar as pesquisas utilizando as características de agilidade mapeadas por Abrantes (2012).

Esta opção foi decorrente da premissa que um determinado gerente de projeto ou gestor de processo ao optar pelo uso de uma determinada prática ágil, implicitamente estará fornecendo às atividades associadas uma determinada característica ou um conjunto de características de agilidade.

Sendo assim, ao iniciar uma pesquisa por relatos de adoção de características de agilidade em atividades de testes esperamos indiretamente recuperar também práticas associadas que estão sendo aplicadas no âmbito destas atividades.

O estudo secundário foi encaminhado utilizando o conjunto das dezoito características de agilidade (dispostas no Apêndice B) e teve como objetivo procurar por estudos e relatos de atividades de testes contemplando estas características, registros de problemas em sua adoção e *trade-offs*. E eventualmente verificar se existem outras características que ainda não tenham sido mapeadas.

O protocolo de revisão sistemática inicial é baseado em (BIOLCHINI *et al.*, 2005), e o método para aplicar a revisão sistemática utilizou a abordagem PICO (PAI *et al.*, 2004). Este procedimento divide a questão de pesquisa em quatro partes: Identificação da População de interesse (*Population*), Intervenção (*Intervention*), Comparação da intervenção (*Comparison*) se aplicável, e resultado (*Outcome*).

Segundo este método, os quatro elementos são conectados através do operador lógico AND e as palavras-chaves que fazem parte do mesmo elemento estarão associadas através do operador OR.

Como o objetivo inicial deste estudo foi caracterizar a adoção das características de agilidade em processos de teste, não foram realizadas comparações, nem foi possível realizar meta-análise. Dessa forma, não foi considerado o elemento de Comparação da Intervenção (*Comparison*), e o estudo neste caso passa a ser uma *quasi-revisão* sistemática (TRAVASSOS *et al.*, 2008).

A questão principal definida foi: Existem relatos de uso de características e/ou práticas ágeis em processos de teste?

E a questão secundária: Existem evidências dos resultados obtidos a partir de sua adoção?

Tendo como problema a ser considerado:

Existem relatos de resultados obtidos na comunidade de desenvolvimento de software com a adoção de características de agilidade em seus projetos. Por outro lado, no contexto de atividades de testes de software é necessário fornecer agilidade

em projetos que apresentam uma quantidade significativa de requisitos funcionais e não funcionais, quando estas atividades podem vir a necessitar de um tempo longo para sua execução.

Na etapa anterior de revisão preliminar da literatura sobre testes e agilidade foram recuperados dois artigos que abordam o conceito de agilidade em processos de testes. Estes artigos foram utilizados como controle para o protocolo da revisão:

- ✓ Artigo 1: Kristian Rautiainen and Maaret Pyhäjärvi, Integrating testing and implementation into development, Engineering Management Journal; Mar 2004; 16, 1; ABI/INFORM Global
- ✓ Artigo 2: Kristian Rautiainen, Juha Itkonen and Maaret Pyhäjärvi, Increasing understanding of the modern testing perspective in software development projects, System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on, 2003.

As fontes de documento foram:

- ✓ Artigos, Relatórios técnicos disponíveis na WEB a serem recuperados através de máquinas de busca às bases de dados eletrônicas.
- ❖ Anais de conferências: Experimental Software Engineering Latin America Workshop- ESELAW (2004 a 2011)

A linguagem considerada nas buscas foi o inglês. As palavras-chaves de cada elemento do protocolo foram:

População:

- Palavras Chave: software application , software system , computer software , software components , project management , software measurement;

Intervenção: Questão Principal: Testes de Software;

- Palavras Chave: testing, validation, verification;

Comparações:

- Não aplicável neste caso

Resultados: Atividades de testes apresentando as características de agilidade.

- Palavras Chave: agility, agile, time boxing, leanness, adaptability, feedback, collaborative, cooperative, people-orientation, incremental, small teams, local teams, interactive, reflection and introspection, transparency, auto-organization , emergence, convergence, modularity, constant testing;

Os critérios de inclusão e exclusão seguem abaixo:

Critérios de inclusão:

- Os documentos devem tratar de características e/ou práticas que visam fornecer agilidade a processos e/ou atividades de teste;

Critérios de exclusão:

- Não serem aplicados a atividades de teste;
- Ao descrever a atividade de testes o artigo deverá fazer referência pelo menos a uma das características ágeis contidas no Apêndice B– sendo que o artigo pode ter utilizado outra terminologia, desde que preserve o conceito associado;
- Os artigos devem descrever a forma proposta de utilização de uma determinada característica na(s) atividade(s) de Teste. Caso contrário, o artigo deverá ser descartado.

A seleção dos estudos foi realizada por dois pesquisadores, sendo um deles com larga experiência em pesquisas nesta área de conhecimento.

O pesquisador de menor experiência realizou uma primeira leitura do título e *abstract* de cada artigo recuperado nas máquinas de busca classificando-os em:

(a) Leitura: Documentos que descrevem atividades de testes observando alguma característica ágil;

(b) Excluído: Documentos que não tratam de características de agilidade em atividades/processos de teste;

(c) Dúvida: Documentos para os quais o pesquisador A ficou em dúvida se deveriam fazer parte da relação de artigos selecionados;

Após esta classificação, o pesquisador com mais experiência leu também o título e *abstract* de cada artigo classificado como Dúvida classificando-os em Leitura ou Excluído.

Na sequência, o pesquisador que fez a primeira seleção leu todos os artigos classificados como Leitura reclassificando-os como: (a) Incluído: Documentos que atendam os critérios de inclusão e não atendam os critérios de exclusão (b) Excluído: Documentos que não atendam os critérios de inclusão ou que atendam os critérios de inclusão e atendam pelo menos um dos critérios de exclusão.

Após a execução do procedimento de seleção foi realizada a extração das informações de interesse do trabalho.

A *string* de busca apresentada abaixo foi devidamente adaptada para as máquinas SCOPUS, WEB OF SCIENCE, IEEExplore e EI Compendex:

("software development" OR "software application" OR "software system" OR "computer software" OR "software components" OR "project management" OR

"software measurement") AND ("testing" OR "validation" OR "verification") AND ((("agility" OR "agile") AND ("time boxing" OR "leanness" OR "adaptability" OR "feedback" OR "collaborative" OR "cooperative" OR "small teams" OR "local teams" OR "incremental" OR "interactive"))

3.1.2 Execução das buscas

A execução das buscas nas bases digitais resultou nas quantidades de referências exibidas no quadro abaixo:

Tabela 3-1 Artigos recuperados na execução do protocolo

Máquina de Busca	Número de artigos encontrados	Controles
SCOPUS	60	1
IEEEExplore	104	1
EI Compendex	54	1
WEB OF SCIENCE	11	0
Duplicados	78	1
Total	151	2

Após a aplicação dos critérios de exclusão e inclusão e da aplicação da estratégia de seleção de estudos com a participação dos dois pesquisadores foram selecionados 13 artigos, sendo que não foi possível recuperar o texto completo de dois destes artigos.

Tabela 3-2 Artigos selecionados após aplicação critérios

Status	Quantidade
Leitura	13
Excluídos	136
Controles	2
Total	151

Os artigos selecionados foram lidos e analisados com o objetivo de verificar sua aderência aos objetivos da pesquisa e as premissas e restrições definidas no protocolo. Nesta etapa foram identificadas a(s) característica(s), prática(s) de agilidade e atividade(s) de testes que cada artigo fazia referência.

Em alguns dos artigos selecionados uma mesma característica estava associada a mais de uma das atividades do processo de teste. Foram encontrados artigos com referência a uma ou mais características, porém sem especificar diretamente uma determinada prática. Contudo, de acordo com o procedimento para seleção, todos os artigos lidos na etapa final faziam menção pelo menos a uma atividade de testes e a uma característica ágil.

Em um percentual substancial dos artigos examinados o alvo dos interesses dos pesquisadores concentrou-se na atividade "Especificação de Casos de Testes", enfocando aspectos como manutenibilidade, capacidade para apoiar a validação dos

requisitos, automatização para facilitar a execução de testes de regressão, e apoio na tomada de decisões de projeto por conta de solicitações de mudanças. Entretanto, ao descreverem esta atividade e os artefatos envolvidos, em nenhum destes artigos foi explicitado o tipo de critério de cobertura definido para seleção e adequação dos casos de Testes (MALDONADO, 2004). O estabelecimento de um critério de cobertura pode apoiar a determinação dos impactos decorrentes de mudanças de escopo, facilitando a realização de estimativas dos riscos envolvidos nas alterações do código.

3.1.3 Análise dos dados

A tabela a seguir apresenta a incidência de atributos de característica de agilidade associadas aos artigos selecionados. Para cada artigo lido, inicialmente foi identificada(s) qual (is) a(s) macro atividade(s) de teste(s) referenciado(s), e para cada macro atividade de testes recuperada qual (is) prática(s) ou adoção de característica(s) eram propostas.

Tabela 3-3 Distribuição das características de agilidade referenciadas

Característica de Agilidade	Quantidade de Artigos	Percentual
Adaptabilidade	5	18,52
Ser Cooperativo	4	14,82
<i>Leanness</i>	4	14,82
Testes Constantes	1	3,71
<i>Time-Boxing</i>	3	11,12
Ser Iterativo	3	11,12
Ser Colaborativo	2	7,41
Ser Incremental	2	7,41
Incorporação <i>FeedBack</i>	2	7,41
Orientação a Pessoas	1	3,71
Total de referencias	27	100

Esta distribuição é exibida no gráfico de barras na figura 3-1. A serie 1 contém as quantidades de artigos que fizeram referência a cada uma das características, e a serie 2 registra o percentual de artigos em relação à quantidade total dos artigos selecionados na etapa de aplicação dos critérios de exclusão e inclusão.

Por vezes ocorreu de um mesmo artigo estar relacionado a mais de uma característica.

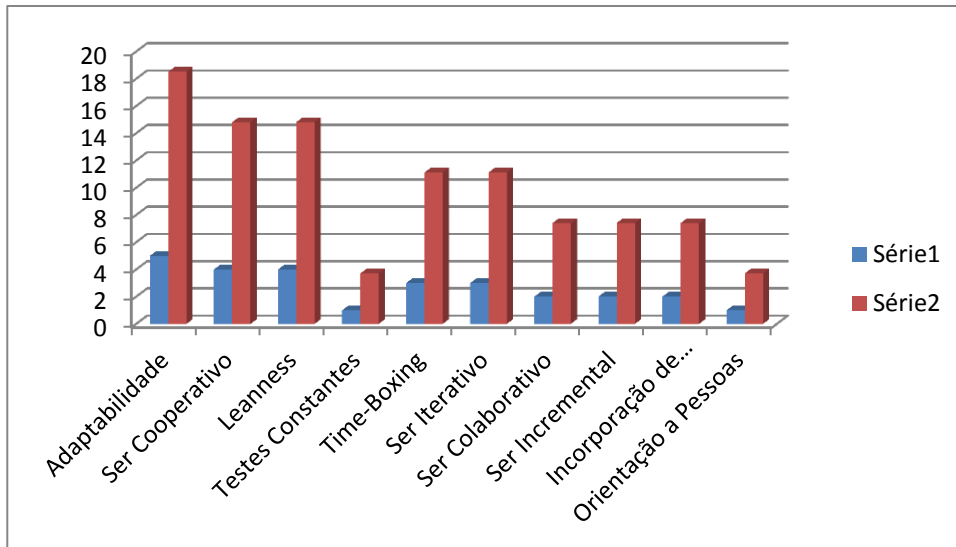


Figura 3-1 – Quantidade de ocorrências de cada característica nos artigos
E um mesmo artigo foi relacionado a mais de uma atividade de Teste:

Tabela 3-4 Distribuição das atividades de testes referenciadas

Macro-atividades de Teste	Quantidade de Artigos
Planejar Teste	3
Projetar Teste	4
Especificar Casos de Teste	10
Definir Procedimentos de Teste	2
Executar Testes	5
Analisar Resultados	0
Monitoramento, Controle e replanejamento de testes	1
Fechamento das atividades	0

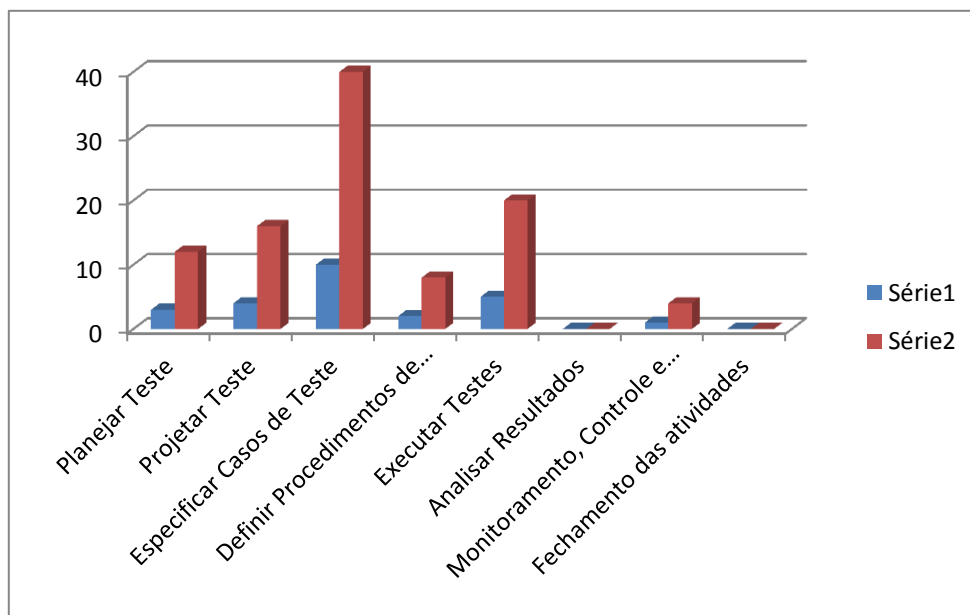


Figura 3-2 - Quantidade de ocorrências de cada atividade testes nos artigos

A macro atividade mais citada pelos artigos foi “Especificar Casos de Teste”. Os estudos relacionados a esta macro atividade são predominantemente sobre manutenibilidade, identificação dos atributos necessários para facilitar a cooperação dos *stakeholders* na definição dos Testes de aceitação, considerações sobre o uso de duplês de testes (*Stub*, *Driver* e *Mocks*) para auxiliar na colaboração de módulos que ainda não foram escritos e automação de Testes de Aceitação.

Em um dos artigos referenciados por um dos trabalhos selecionados é apontada a necessidade de adoção de uma linguagem comum (ubíqua), com termos bem definidos do domínio do negócio, a ser utilizada por todas as pessoas que fazem parte do processo de desenvolvimento de software [ANDREA, 2009].

Muitos destes trabalhos são estudos sobre a abordagem de desenvolvimento TDD (*Test-Driven Development*).

A segunda macro atividade que aparece com mais trabalhos associados é “Executar Testes”, seja por conta do enfoque à integração entre os processos de desenvolvimento e testes e da necessidade de um rápido *feedback* sobre incidência de defeitos, como também decorrente da necessidade de automação desta etapa do processo de teste.

Duas das ocorrências da macro Atividade “Projetar Testes” são derivadas dos artigos de controle quando estes promovem a adoção de um *framework* baseado no modelo V. As outras duas ocorrências são decorrentes respectivamente: (a) de um artigo sobre desenvolvimento baseado em Testes de Aceitação de Estórias de Usuário em que a atividade “Priorizar Casos e Procedimentos de Teste” é referenciada indiretamente através da prática “Jogos de planejamento” (b) do artigo “*Herding Cats*” [KESSLER, 2009] em que são relatadas algumas dificuldades na implantação de estratégias de testes.

Apenas o artigo selecionado “*A control approach for Agile Approach*” [CANGUSSU, 2005] abordou a macro atividade de “Monitoramento, Controle e replanejamento de testes”, através da proposição de um modelo de inferência para acompanhamento de um processo de desenvolvimento baseado no paradigma ágil.

As macro atividades “Analisar Resultados” e “Fechamento das atividades” não foram citadas em nenhum artigo.

Parte deste comportamento pode ser decorrente de duas evidências: (a) apenas em dois artigos do protocolo (os artigos de controle) são feitas referências explícitas a um processo formal de Teste, com suas atividades de planejamento e controle de testes (b) as duas atividades com alto dispêndio de esforço associado são respectivamente “Especificar Casos de Teste” e “Executar Testes”.

No cruzamento entre as atividades de testes e as características de agilidade, percebe-se que ao estudar a atividade de testes “Especificar Casos de Teste” a características com mais associações estabelecidas é “Adaptabilidade”. Isto porque a existência de um conjunto de casos de Testes cujo desenvolvimento foi baseado em um critério de cobertura pré-definido, e com rastreabilidade aos requisitos do sistema, apoia a gerência de mudanças, na medida em que permite visualizar os riscos associados no atendimento às alterações adaptativas e evolutivas.

A distribuição dos 13 artigos utilizados neste estudo, por ano de publicação ficou conforme descrito na tabela abaixo:

Tabela 3-5 Distribuição dos artigos por ano de publicação

Ano de Publicação	Quantidade de artigos
2002	1
2004	1
2005	1
2007	2
2008	1
2009	6
2010	1

Quanto aos seis artigos escritos em 2009: três estudos sobre Testes de Aceitação, uma comparação entre *Test-First* e *Test-Last*, um relato de adoção de Integração Continua, e uma proposta de uso de uma abordagem de desenvolvimento dirigida a Testes de Aceitação.

Os dois artigos escritos em 2007 são respectivamente sobre testes de desempenho e uso de diagramas de cenário para apoiar a visualização de Casos de Testes de Aceitação.

Não foram encontrados desdobramentos dos trabalhos relatados nos dois artigos do grupo de controle sobre uma proposta de *framework* considerando os atributos “Ser Iterativo” e “*Time-Boxing*”.

O artigo mais recente, escrito em 2010 é sobre o uso de Testes de Aceitação para validação de requisitos. O que pode vir a configurar uma tendência para desenvolvimento de trabalhos associados a testes de aceitação.

De qualquer forma, a baixa quantidade de artigos recuperados não é representativa para que possamos realizar projeções.

3.1.4 Ameaças à Validade

Na etapa de definição das palavras-chaves, após a realização de algumas simulações com o intuito de minimizar a recuperação de uma grande quantidade de itens de baixa relevância, foram incluídas as palavras *agile* e *agility* para formação da *string* de Busca.

Após esta alteração foi constatado que muitos artigos relacionados ao tema continuavam sendo recuperados. Contudo pode ser que a sensibilidade tenha diminuído, e algumas referências relevantes não tenham sido incluídas.

Considerando a crescente publicação de artigos e ensaios na WEB, é possível que alguns trabalhos com conteúdo rico sobre o tema, e que não estão indexados nas bases de busca, não tenham sido recuperados.

Na primeira leitura os dois pesquisadores aplicaram os critérios de inclusão e exclusão após a leitura dos abstracts. Como nem sempre os abstracts descrevem exatamente o conteúdo do estudo, pode ser que algum experimento relevante sobre o tema tenha sido desclassificado.

A classificação de cada artigo em relação às atividades de testes e características de agilidade foi realizada a partir de critérios subjetivos dos pesquisadores envolvidos. A quantidade pequena de artigos retornados também pode ser um viés para a análise dos dados quanto à categorização dos artigos.

Apesar das ameaças acima, ao comparar os resultados da execução do protocolo com estudos secundários em temas correlatos (DYBA & DYNGSOR, 2009) e outras revisões na literatura técnica (CONBOY, 2009) é confirmada a escassez de relatos de estudos de caso sobre adoção de características ágeis em processos de software em geral, incluindo neste caso as atividades de teste.

3.1.5 Perspectivas Futuras

3.1.5.1 Estado atual da pesquisa e Resultados Preliminares

Ao buscar uma síntese do estado da arte do uso de características ágeis em processos de testes não foram recuperados trabalhos desenvolvidos com o propósito de identificar, e experimentar quais destas características poderiam ser associadas às atividades de um processo de testes pré-definido.

Os artigos selecionados fazem menção a atividades do processo de testes referenciadas na norma IEEE-829. Porém em cada um destes estudos e relatos, a atividade de testes não estava sendo conduzida a partir da especificação de um processo de testes padrão definida previamente.

Com exceção dos estudos relatados nos artigos de controle que esboçam uma crítica ao modelo V, não foram encontrados relatos de uso de processos de testes com diretrizes específicas para apoiar a adoção de características e práticas ágeis.

Em alguns dos trabalhos recuperados existem indícios de ganho de produtividade e qualidade na execução de atividades de testes com adoção de práticas e características como descrito nos artigos, (DOBSON, 2007), (HAUGSET &

HANSSEN, 2008), (KESSLER & ANDERSEN, 2009), (HAUGSET & HANSSEN, 2009), (CONNOLY et al, 2009), (STOLBERG, 2009).

Apesar de dois artigos selecionados (PYHÄJÄRVI et al, 2003), (GALLARDO-VALENCIA & SIM, 2010) abordarem a necessidade de definição de uma estratégia baseada em riscos, para priorização do projeto e execução dos Casos de Testes, não é detalhado como esta estratégia deve ser adotada.

Esta carência de definição formal dos critérios, atividades e estratégias aparece de forma mais clara em (HANSSEN, 2009) quando é descrito um estudo de caso na indústria, realizado com o objetivo de observar os resultados de adoção de uma estratégia baseada em Testes de aceitação automatizados. Ao final do estudo os pesquisadores incluíram na lista de experiências negativas e problemas, a falta de critérios de testes na seleção de Casos de Testes. Quando novos testadores sem conhecimento do domínio do problema eram incorporados ao time, a quantidade de Casos de Testes a serem executados crescia substancialmente. Os pesquisadores concluem que a abordagem de desenvolvimento baseada em Casos de Testes de Aceitação deve ser apoiada por um processo de trabalho apropriado ou modelo. Uma ferramenta por si só pode não ser suficiente (HAUGSET e HANSSEN, 2008).

Neste estudo de Haugset e Hanssen é feita referência a outro artigo sobre casos de testes funcionais e aceitação no qual é apontado um conjunto de características a serem consideradas na atividade de “Especificar Casos de Teste” quando a técnica de cobertura é funcional: (a) Facilidade para escrita - Se eles se tornam gargalo para a produção de código então serão considerados opcionais e serão incompletos e obsoletos. (b) Confiabilidade – Erros nos testes irão mascarar defeitos no código produzido ou ainda gerar falsos positivos. (c) Facilidade de Leitura – O pessoal do negócio deve ter facilidade para checar a completude e correção da regra de negócio/requisito especificado. (d) Manutenibilidade – Os Testes Funcionais não possuem a rede segura de Testes de Regressão que o código possui. (e) Facilidade para Localização – Os testes funcionais devem ser localizados e atualizados antes do novo código ser colocado em produção, por isso devem ser facilmente identificados, recuperados e controlados a partir de um processo de gerência de configuração (ANDREA, 2007).

3.1.5.2 Adoção de Características Ágeis em um Processo de Teste

Em “Increasing Understanding Of The Modern Perspectives” (PYHÄJÄRVI, 2003) ao ser formulada uma crítica à adoção do modelo V observa-se que em um ciclo de vida cujo desenvolvimento é iterativo, a definição dos Casos de Testes deve focar em requisitos cujo conhecimento já esteja suficientemente maduro. Entretanto é preciso

distinguir alguns casos em que a falta de entendimento dos requisitos não é decorrente da pouca maturidade do domínio, ou do fato do projeto estar em suas etapas preliminares, mas essencialmente pela ausência de boas práticas de elicitação e análise, e técnicas de inspeção dos documentos do projeto, quando a equipe não é capaz de especificar o requisito e validar sua consistência nas várias perspectivas i.e. projetista, testador, analista de sistemas.

Em um artigo complementar de 2004 “Integrating Testing and Implementation into Development” (PYHÄJÄRVI, 2004) Rautinein e Pyhajarvi sinalizam para a pouca existência de trabalhos focando a integração entre o ciclo de desenvolvimento de software e processo de teste, e destacam a importância do planejamento das atividades de testes e recursos envolvidos, de forma a suportar os ciclos de interação entre os processos de desenvolvimento e testes.

Alguns trabalhos sobre o modelo V apontam que em determinados projetos pode ser necessário rever as especificações dos planos de testes, por conta da evolução do conhecimento do domínio ou mesmo por pequenas mudanças realizadas em nível de projeto.

Em um dos artigos de controle, Rautinein (2004) destaca crítica ao V Model elaborada por Marick (1999) quando é sugerido que o modelo a ser adotado apresente uma atividade específica de análise de defeitos encontrados, visto que a partir desta análise é possível apontar prováveis ocorrências de falhas em outras partes do sistema, sinalizando assim a necessidade de execução de outros testes.

A ideia é prover ao processo de testes a característica ágil de *feedback* contínuo. Entretanto Marick (1999) não chega a desenvolver como esta atividade poderia ser implementada. No mesmo artigo é apontada a necessidade de que o processo de testes apoie e facilite a manutenção das especificações dos Casos de Testes à medida que novas informações sobre o modelo sejam adquiridas pela equipe do Projeto.

No artigo “A control approach to agile process” sobre modelo de predição de defeitos Cangussu (2008) sugere o aproveitamento da capacidade de rápido *feedback* em relação aos defeitos encontrados na execução de ciclos curtos e incrementais, para propor um *framework* capaz de detectar as possíveis fases de um processo de desenvolvimento que apresentam necessidade de melhorias. Este *framework* utiliza-se do modelo ODC proposto por Chillarege (1992).

Em relação aos desafios e as tarefas associadas à manutenção de um conjunto extenso de Casos de Testes, Kessler e Andersen sugerem a definição de critérios distintos para a escolha dos dados a serem utilizados nos testes de unidade e aceitação (KESSLER e ANDERSEN, 2009). Neste relato é sugerido o uso de objetos

Mock para emulação de comportamento dos dados de produção sem que seja necessário utilizar massa real na execução dos Casos de Testes, o que poderia gerar dependência desnecessária (FOWLER, 2010). Este mesmo recurso é sugerido para minimizar eventuais dificuldades na realização de testes a sistemas externos.

Em projetos para os quais foi escolhida a técnica de desenvolvimento de casos de Testes automatizados, assegurar uma boa manutenibilidade para estes artefatos - através do uso de boas práticas que evitem acoplamento e solucionem dependências externas - significa dotar a atividade de especificação de Casos de Testes a característica adaptabilidade.

Em muitos destes relatos, são descritas técnicas aplicadas em projetos que utilizam métodos ágeis cujas atividades de testes são executadas a partir de uma abordagem *ad hoc*, sustentada por técnicas emergentes - como é o uso de objetos *Mock* e Testes automatizados - sem considerar a possibilidade do uso conjunto destas técnicas com estratégias já experimentadas pela comunidade de testes como a identificação de riscos na execução das atividades de Projeto de Casos de Teste, em um processo formal como o proposto por Dias e Travassos, um processo derivado da norma IEEE-829 (DIAS NETO e TRAVASSOS, 2006).

Isto porque a dispensa de um conjunto de diretrizes e estratégias de testes com resultados já comprovados de sucesso, na academia e na indústria, pode ser crítico, independentemente do tamanho do projeto a ser empreendido.

3.1.5.3 Tema de Pesquisa

No livro “*Balancing Agility and Discipline*” (BOEHM e TURNER, 2004), os autores concluem que os Métodos Ágeis são capazes de tratar as mudanças através da construção de uma visão dos objetivos e estratégias do projeto, e do compartilhamento de conhecimento tácito entre os vários membros da equipe. Porém não são capazes de escalar suas práticas para grandes e complexos projetos, por não promoverem conformidade para aspectos críticos como especificação de interfaces e arquiteturas de linha de produto.

Por outro lado, projetos orientados a planos oferecem conformidade através da externalização do conhecimento, com forte ênfase no uso de documentação, carecendo, todavia de adaptabilidade.

Entretanto, poucos trabalhos experimentais contribuem para geração de evidências sobre custos, prazos e outros elementos de projeto na adoção de práticas ágeis em relação às obtidas no uso de uma abordagem orientada a planejamento. Este cenário direciona para oportunidades de pesquisas a serem encaminhadas em

relação às atividades de testes em organizações que possuem um processo de teste formal.

Contribuindo para a proposição de um processo de testes com capacidade de adaptação às mudanças, transparência, visibilidade de projeto e outros atributos visando ganhos de produtividade.

Nos estudos selecionados, recuperamos relatos de esforço dispendido especificamente para as atividades “Especificar Casos de Testes” e “Executar Casos de Teste” sugerindo que estas áreas poderiam ser as mais beneficiadas através da utilização de técnicas e práticas que fornecessem agilidade na sua execução. Considerando as dificuldades relatadas nestes trabalhos, será conduzido um estudo de caso na indústria, com o objetivo de aplicar algumas das características e práticas ágeis, mapeadas por Abrantes, nas atividades de um processo de testes já existente (ABRANTES, 2012).

3.2 Conclusão

Os resultados da revisão inicial da literatura e da execução do protocolo de *quasi*-revisão sistemática descrevem alguns dos esforços de pesquisa em Engenharia de software, no sentido de prover maior agilidade às atividades de projeto e execução dos testes.

No âmbito da indústria, na implementação de uma tecnologia ou método, os responsáveis pela condução de projetos de teste de software muitas vezes são obrigados a tomar decisões de qual opção é a mais recomendada para seu projeto. Porém, sem ter à sua disposição um conjunto de evidências à respeito de quais são os limites, custos, riscos e compatibilidade destes recursos para implementá-las em um determinado projeto.

Dyba et al (2005) ao descreverem a importância da condução de um conjunto de atividades de engenharia de software, baseada em evidências (EBSE- *Evidence Based Software Engineering*), sugere a execução de uma sequência de passos para o profissional que se depara com a necessidade de escolher uma determinada tecnologia ou método:

1. Converter um problema em uma pergunta a ser respondida.
2. Pesquisar na literatura em busca de evidências capazes de respondê-la.
3. Avaliar criticamente as evidências de sua validade, impacto e aplicabilidade.
4. Integrar as provas avaliadas com as práticas, experiência e valores do cliente, bem como as circunstâncias a serem levadas ao se tomar as decisões de como utilizá-la.

5. Avaliar o desempenho e buscar formas de melhorá-lo.

Sendo que, muitas das atividades necessárias para a aplicação de uma engenharia de software baseada em evidências já foram desenvolvidas e sistematizadas nas atividades de um processo de melhoria de software:

1. Identificar um problema.
2. Propor uma tecnologia ou procedimento para resolver esse problema.
3. Avaliar a tecnologia proposta em um piloto projeto.
4. Se a tecnologia é adequada, adota-la e implementá-la.
5. Monitorar a organização após a implementação da nova tecnologia.
6. Retornar ao passo 1.

Considerando estas abordagens, e a execução dos itens 1 e 2 propostos em uma estratégia ESBE, no próximo capítulo será descrita a proposição de um conjunto de procedimentos, visando fornecer a um processo de testes um subconjunto de características de agilidade.

Para isto o processo de testes da organização será avaliado em termos de aderência a alguns padrões sugeridos na norma IEEE-829, e também às eventuais carências de agilidade do processo, relatadas pela organização.

4 Procedimento para Introdução de Práticas ágeis em Atividades de Teste de Software

Este capítulo apresenta um procedimento projetado para composição de uma estratégia de introdução de práticas ágeis em um processo de teste, a ser executado em campo, e constituído das seguintes atividades: (a) Elicitação e documentação das atividades de testes do processo de desenvolvimento de software de uma organização, (b) Comparação das atividades de testes da organização com atividades de um processo especializado a partir da norma IEEE-829, bem como análise dos artefatos utilizados (c) Seleção de oportunidades de inserção de práticas de mapeamento entre práticas ágeis e atividades de testes recuperadas na fase de revisão da literatura técnica. (d) Proposição de um estudo de caso a partir das observações coletadas nas etapas anteriores.

4.1 Introdução.

A gestão da mudança organizacional é um elemento essencial do processo de *software* para o sucesso dos esforços de melhoria. As mudanças em processos devem ser planejadas, monitoradas e medidas para que seja possível avaliar seu impacto.

Runeson e Host (2009) ao caracterizarem um estudo de caso como uma metodologia adequada para pesquisa em engenharia de *software*, por estudar o fenômeno contemporâneo em seu contexto natural, registram também sua utilização em estudos de transferência de tecnologias (GORSCHK, 2006) e em melhoria de processos de *software* (DITTRICH, 2008).

De acordo com estas premissas, o objetivo principal na etapa de definição dos procedimentos de preparação para elaboração do estudo de caso é compreender e avaliar o processo de testes de *software* na organização utilizada na abordagem proposta, para que seja possível propor mudanças a fim de contribuir para a melhoria deste processo. Neste contexto, a proposição de práticas ágeis nas atividades de testes e os ajustes decorrentes podem ser mais apreendidos se acompanhadas por um processo de melhoria de *software*.

Alguns estudos sinalizam que para obter sucesso em iniciativas de melhoria, além de ser necessário ter experiência técnica na condução destas atividades, é

necessário também ter conhecimento sobre a cultura da organização para que a implementação seja adequada e efetiva (SCHOTS *et al*, 2011).

Sob este ponto de vista, um dos pré-requisitos para assegurar uma razoável aderência às necessidades da organização, é a participação ativa de membros da empresa motivada em compartilhar algumas características do processo de *software* corrente com o engenheiro responsável pela introdução das práticas ágeis.

Quando a organização selecionada possui um processo de melhoria de *software* implantado, em regime de execução, os resultados deste trabalho podem mostrados através de duas perspectivas: (a) da pesquisa realizada, a partir dos resultados do estudo de caso; (b) e das atividades organizacionais de melhoria de processo da empresa.

Aoyama (1998) aponta como um dos fatores de sucesso na implementação das práticas ágeis, a presença na organização de um processo de *software* definido, apresentando um nível de maturidade. De acordo com este estudo, é razoável esperar uma maior facilidade na inserção de agilidade, em organizações cujos processos apresentem um maior nível de maturidade. No entanto, o pré-requisito para a utilização desta abordagem é que tenha um processo de testes definido e executado na organização no âmbito dos projetos de software.

4.2 Procedimento para Inserção de Características de Agilidade

Um processo de desenvolvimento de *software* deve ser transparente o suficiente para que seja possível acompanhar sua execução, permitindo observar e avaliar os artefatos produzidos. Tais artefatos, por exemplo, casos de uso, protótipos intermediários, casos de teste, dentre outros, poderão ser utilizados para apoiar uma avaliação do processo atual. Como resultado desta avaliação, é possível decidir entre prosseguir para o passo seguinte ou reiterar a etapa anterior (CUGOLA e GHEZZI, 1998).

Relatos das implantações de melhorias de processo de *software* mostram um ganho substancial em qualidade, produtividade e redução do tempo de ciclo de desenvolvimento (*cycle-time*). Em alguns destes relatos, com a adoção de um programa de melhoria baseado no CMM, é descrito uma redução em oito vezes do *cycle-time*, se comparado ao resultado de um novo projeto a *baselines* de projetos similares, com mesmo domínio de problema (DIAZ e SLIGO, 1997).

Com o objetivo de obter melhorias no processo de testes da organização, é proposta uma sequência de procedimentos na intenção de implantar características de agilidade em suas atividades de projeto, construção e execução de testes.

4.2.1 Etapas da Abordagem.

A abordagem proposta utiliza as mesmas etapas definidas no paradigma de melhoria, contínua apresentado no *framework* conceitual Fábrica de Experiência proposto por Basili (BASILI *et al*, 1992): (a) Caracterização: fase na qual deve ser obtido compreensão sobre o contexto que envolve o desenvolvimento de *software*, a partir dos modelos disponíveis, os dados, e outros elementos de modo que similaridades em relação a outros contextos foram reconhecidas. (b) Planejamento: baseando-se na caracterização realizada na fase anterior, são definidos os processos adequados, ferramentas e métodos para alcançar os objetivos de qualidade definidos. (c) Execução: etapa em que os processos são executados, e na qual deverá ser fornecido feedback em tempo real durante a construção dos produtos. (d) Empacotamento: Ao final de cada projeto é executada a análise dos dados e das informações coletadas, para avaliação das práticas em curso, determinação dos problemas e registro do conhecimento criado e adquirido, permitindo o registro de recomendações para melhorias futuras. A experiência adquirida está armazenada através da atualização dos modelos que descrevem o processo de desenvolvimento, bem como outras formas de conhecimento.

Utilizando este *framework*, um conjunto de procedimentos será proposto para inserção de práticas ágeis em um processo de testes de *software*. Na figura 4-1 é exibido um diagrama contendo estas atividades, descritas nos próximos tópicos. Estes procedimentos foram modelados no EPF (*Eclipse Process Framework Composer*), e publicados em <http://lens-ese.cos.ufrj.br/insercaoagil/>, disponíveis para os participantes do estudo. Algumas das atividades dispostas no diagrama foram executadas conjuntamente pelo engenheiro de *software* e pela equipe responsável pela melhoria das atividades na organização.

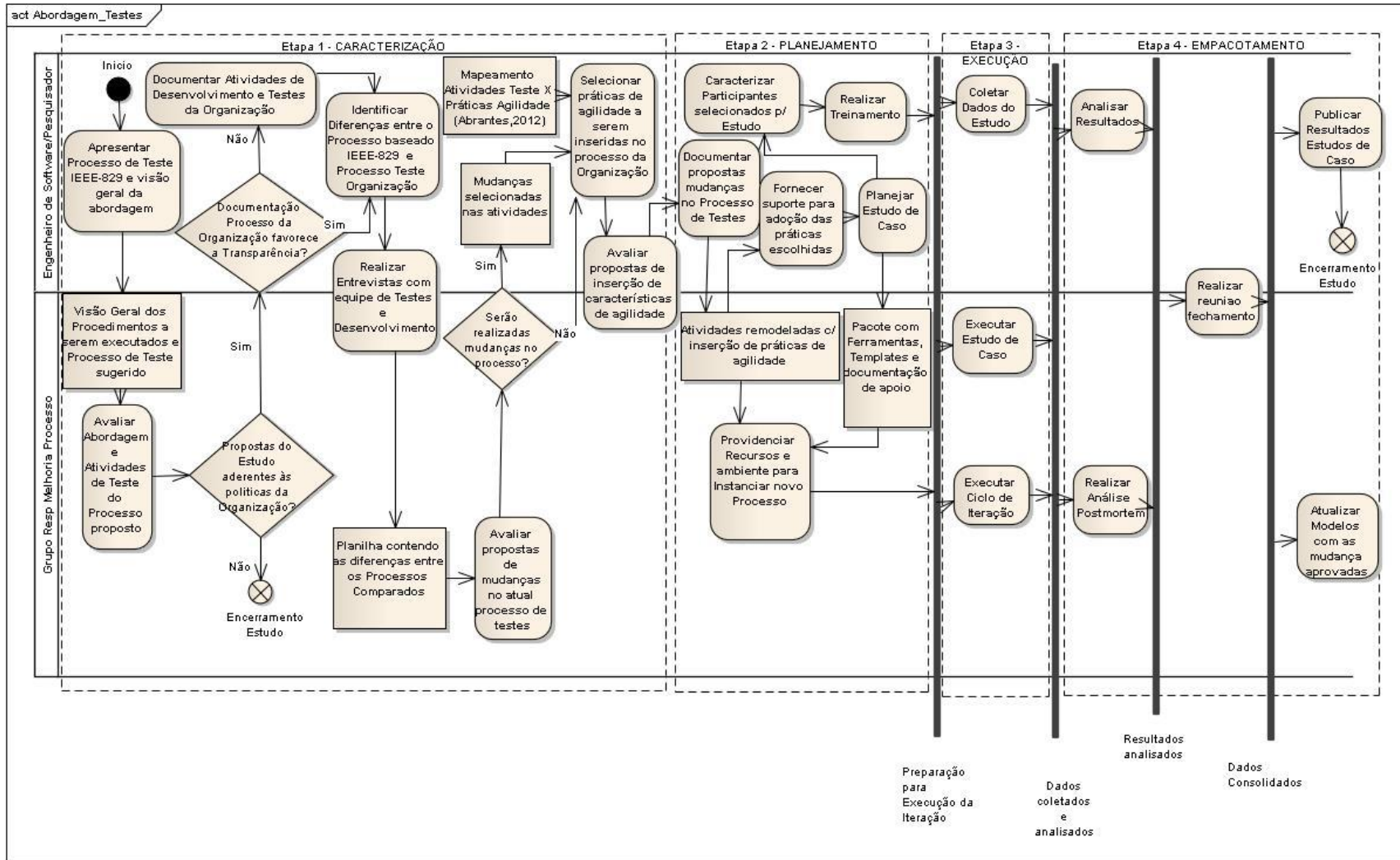


Figura 4-1 - Abordagem para Inserção de Características de Agilidade em Processos de Testes

4.2.1.1 Etapa 1 - Caracterização

Na primeira fase, o pesquisador descreve uma visão geral da proposta de trabalho, apresentando-a para a organização que deseja obter melhorias em seu processo de testes de *software* e dotá-lo de características de agilidade (descritas no Apêndice B). Nesta visão, estão contidos os procedimentos a serem executados no decorrer do estudo, e os instrumentos utilizados.

Após esta apresentação, os envolvidos avaliam a aderência da abordagem sugerida às políticas da organização, examinando o conjunto das atividades propostas, o processo de teste a ser utilizado como padrão e os esforços dispendidos para a consecução destas atividades.

Estando a organização de acordo com a proposta de trabalho, o engenheiro de *software* fará um levantamento da documentação existente sobre os processos de teste e desenvolvimento, e avaliará se a forma de documentação das atividades de teste e desenvolvimento favorece o compartilhamento das informações sobre responsáveis, artefatos gerados, práticas utilizadas. Sendo capazes de refletir rapidamente mudanças no processo, facilitar o acesso aos *templates*, *guidelines* e modelos utilizados - de acordo com a prática “Visibilidade de Projeto” (descrita no Apêndice D).

Estes pré-requisitos convergem com algumas das diretrizes sugeridas por HOLTZ (2001) para a gestão de conhecimento de um processo, cujos modelos de processo não servem apenas como um meio de armazenamento sobre melhores práticas, mas também provêm links para itens de conhecimento.

Para os casos em que as atividades de teste juntamente com as atividades de desenvolvimento não estejam devidamente documentadas, poderão ser sugeridas formas de documentação que promovam o compartilhamento das informações do processo entre os praticantes, favorecendo atualização rápida e frequente quando necessário.

Independente da linguagem escolhida, esta deverá ser aderente às diretrizes contidas na prática de agilidade “Visibilidade de Projeto”, capaz de promover a criação de uma visão comum dos processos, e, ao mesmo tempo, facilitar o entendimento das mudanças que serão sugeridas.

Após a documentação do processo de desenvolvimento e testes da organização deverá ser executada uma comparação entre as atividades de testes correntes e as atividades de teste do processo utilizado como padrão e especializado a partir da norma IEEE-829 (Apêndice C).

Para isto, foi definida uma taxonomia que considera os vários resultados na comparação entre as atividades de cada um dos processos (Tabela 4-1).

Como nesta comparação, podem ocorrer situações em que uma determinada atividade do processo de testes (utilizado como padrão) não está contemplada por nenhuma atividade do processo de testes da organização, como também a situação inversa - isto é, o processo de testes da organização incluir atividades que o processo padrão não contempla - foi necessário definir um conjunto de classificadores que atendessem às possíveis situações decorrentes desta comparação.

Tabela 4-1 Taxonomia para apoiar comparação entre atividades dos Processos ³.

Classificação		Critério
QUE	Atividade equivalente	A Atividade do Processo de Testes implantado na empresa é compatível com a definida no Processo Padrão.
EQUt<>	Atividade equivalente e realizada em etapa distinta	A Atividade do Processo de Testes implantado na empresa é compatível com a definida no Processo Padrão, porém ocorre em etapa anterior ou posterior.
EQU-	Atividade equivalente com complementação	Na relação das tarefas, artefatos e perfis definidos na atividade do Processo de Testes da empresa falta(m) um ou mais itens, presentes no Processo Padrão. (*)
EQU-t<>	Atividade equivalente com complementação e realizada em etapa distinta	Na relação das tarefas, artefatos e perfis definidos na atividade do Processo de Testes da empresa falta(m) um ou mais itens, presentes no Processo Padrão. Além disso, a atividade é realizada em etapa distinta da etapa em que é proposto no Processo Padrão.
EQU+	Atividade equivalente com elementos adicionais	Existe(m) tarefa(s) e/ou artefato(s) e/ou perfil(s) definido(s) na atividade do Processo de Testes da empresa e que não está (ão) definido(s) na atividade associada no Processo Padrão. (*)
NEQ	Não Equivalente	Não existe atividade no Processo de Testes da empresa equivalente a esta atividade do Processo Padrão
INE	Inexistente	Não existe atividade no Processo Padrão equivalente a esta atividade do Processo de Testes da empresa
(*) Ao comparar duas atividades que apresentam objetivos e propósitos semelhantes poderá ocorrer de uma atividade do processo da empresa avaliada apresentar uma tarefa que não existe no processo padrão, e ao mesmo tempo, o processo padrão conter outra tarefa, perfil ou artefato que não está presente no processo implementado pela empresa. Neste caso ao classificar estas atividades serão utilizados simultaneamente os mnemônicos EQU+ e EQU-.		

Além disso, foi constatada a necessidade de levar em conta a dimensão tempo, isto é, atividades comuns, com mesma descrição e objetivo, mas que estivessem modeladas para ocorrer em etapas distintas de cada um dos processos estudados.

³ Adaptação planilha proposta por Mello (2011).

Para facilitar a comparação entre as atividades de cada processo foi utilizada uma planilha, na qual o pesquisador deverá registrar as diferenças entre os processos (Apêndice E).

Nesta comparação, além das descrições de cada uma das atividades serão também comparados os artefatos de entrada e saída.

Após este mapeamento, o pesquisador deverá realizar as entrevistas com a equipe de testes e equipe responsável pela melhoria de processos, de forma a complementar as informações obtidas pela comparação realizada entre os processos.

Com os dados consolidados, realizam a reunião com os membros da organização (equipe de testes e equipe responsável pela melhoria de processos) para que sejam definidas as mudanças no processo de testes atual da organização.

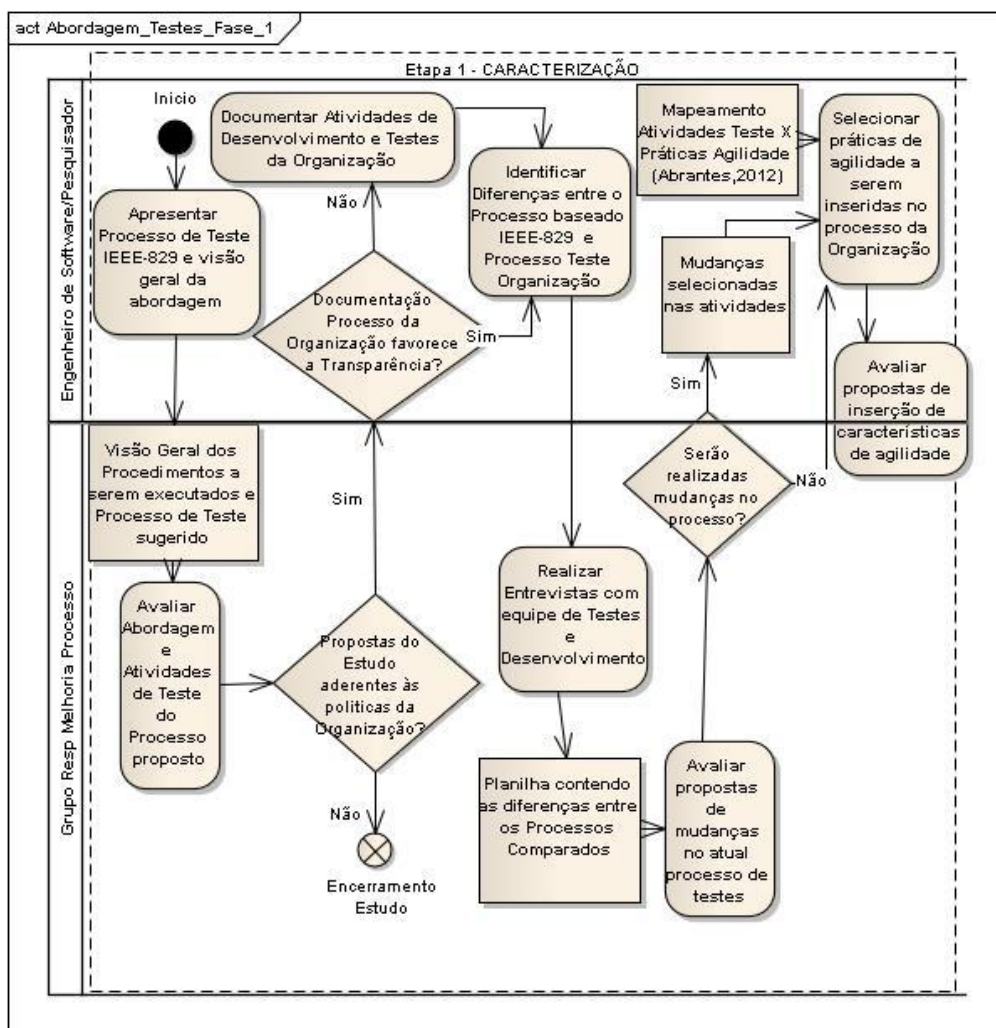


Figura 4-2 – Procedimentos da Etapa 1 de Caracterização

Na etapa de avaliação, pode-se concluir em não considerar nenhuma das diferenças apontadas pela planilha e, por conseguinte, não implementar nenhuma mudança no processo nesta etapa. Por exemplo, é possível que os *gaps* existentes não sejam determinantes para eventuais melhorias no processo atual, devido às

características de operação da organização (domínio de problema, criticidade do ambiente e outros). Contudo, uma avaliação cuidadosa deverá ser feita nesta etapa para que a ausência de uma determinada atividade não venha desfavorecer o alcance da almejada agilidade perseguida através da adoção de uma determinada prática.

Como no mapeamento proposto por Abrantes (2012) é associado a cada atividade do processo de teste padrão um conjunto de práticas ágeis, pois a ausência de uma destas atividades ou sua execução parcial poderá minimizar o efeito de adoção das práticas selecionadas conjuntamente pela organização e o pesquisador.

Essa seleção utiliza o mapeamento descrito no Apêndice D e deve ser realizada após a avaliação das diferenças apontadas nos processos. Assim, é possível determinar melhores oportunidades de uso das praticas ágeis.

Após esta escolha, o pesquisador obtém a relação total de mudanças que serão realizadas no processo, a fim de que estas mudanças sejam devidamente documentadas em uma linguagem definida para modelagem do processo na próxima fase.

4.2.1.2 Etapa 2 - Planejamento

Nesta etapa, as mudanças consideradas são documentadas em uma nova versão do processo de testes e desenvolvimento da organização, e o engenheiro de software projeta e fornece o material de suporte necessário para a implementação. Este material de suporte é fornecido na forma de um conjunto de diretrizes, na indicação de uma determinada técnica de teste ou numa ferramenta de apoio para a realização das atividades.

O projeto deste material visa apoiar a organização na adoção da prática sugerida e devem ser divulgados através de um treinamento presencial, quando serão avaliados o entendimento dos participantes em relação aos recursos dispostos e os objetivos a serem alcançados. Este treinamento faz parte do conjunto de eventos planejados no estudo de caso conduzido na organização com o objetivo de observar o fenômeno de inserção destas práticas nas atividades de teste selecionadas.

A organização fornece os recursos (ambiente, equipamento, sala de treinamento) para viabilizar a realização do estudo.

Também deve ser selecionado um projeto a se utilizar do novo processo proposto.

Para os casos em que a organização planeja e implementa os processos necessários de monitoramento, medição, análise para demonstrar conformidade do produto deverão ser examinadas as métricas relacionadas às atividades de verificação, validação e testes. Estas métricas conjuntamente com os dados obtidos

na execução do estudo de caso estarão apoiando a avaliação dos resultados da inserção de práticas ágeis nos processos de testes.

Preferencialmente, escolher um ciclo de desenvolvimento que não requeira um acompanhamento demasiadamente prolongado facilitando a observação dos resultados das mudanças, suas avaliações, ajustes e replanejamentos. O planejamento do estudo de caso, contemplando o treinamento, eventos propostos e o contexto de execução estão descritos no capítulo 5.

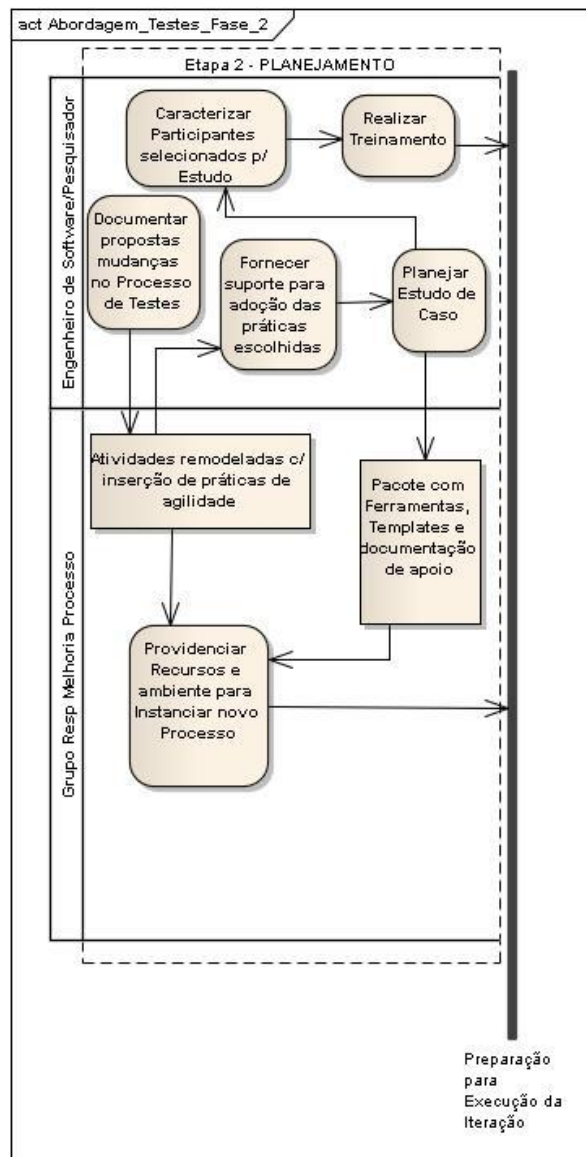


Figura 4-3 - Etapa 2 da Abordagem Proposta – Planejamento

4.2.1.3 Etapa 3 - Execução

Nesta etapa, será executado o estudo de caso paralelamente à condução das atividades de desenvolvimento e teste do ciclo de desenvolvimento selecionado, o qual utiliza uma versão do processo com as mudanças propostas.

Os dados do estudo de caso foram coletados pelo pesquisador.

Ao final da execução, os participantes receberão um questionário para a avaliação qualitativa dos resultados obtidos, e coletado os dados relacionados às variáveis dependentes que se referem às hipóteses formuladas para apoiar a avaliação sobre as vantagens e desvantagens de incorporação das mudanças sugeridas. Esta avaliação aparece na próxima etapa de empacotamento.

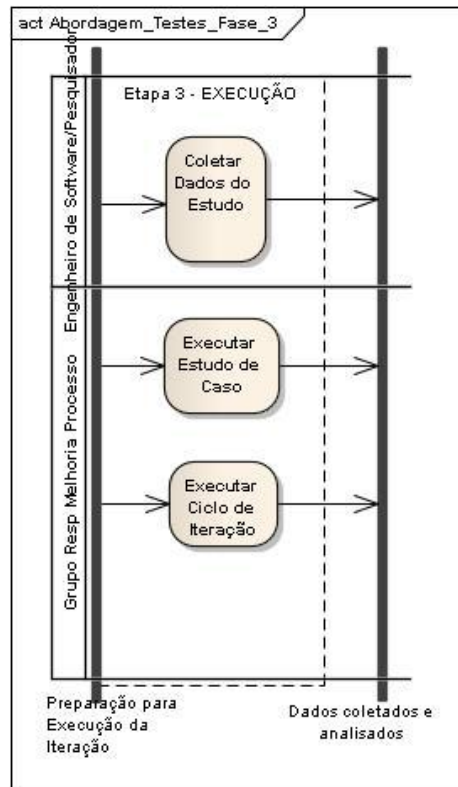


Figura 4-4 - Etapa 3 Execução

Os resultados obtidos na execução desta etapa estão descritos no capítulo 5, juntamente com os resultados da etapa de Empacotamento.

4.2.1.4 Etapa 4 - Empacotamento

Nesta etapa, o pesquisador analisa os resultados coletados durante a execução do estudo de caso, realiza reunião com os participantes do estudo para complementar alguma informação nos questionários de avaliação distribuídos ao final da execução e verifica as hipóteses para publicação dos resultados.

O gerente do projeto piloto observa os resultados obtidos a partir das mudanças propostas, de posse das informações geradas na reunião de *post-mortem*, atualiza o modelo com eventuais modificações que se façam necessárias para refletir as mudanças no processo, utilizando a versão do modelo desenvolvida na etapa 2 de Planejamento, que abrange as propostas de mudança sugeridas para o processo.

A figura 4-5 mostra as atividades a serem executadas nesta etapa.

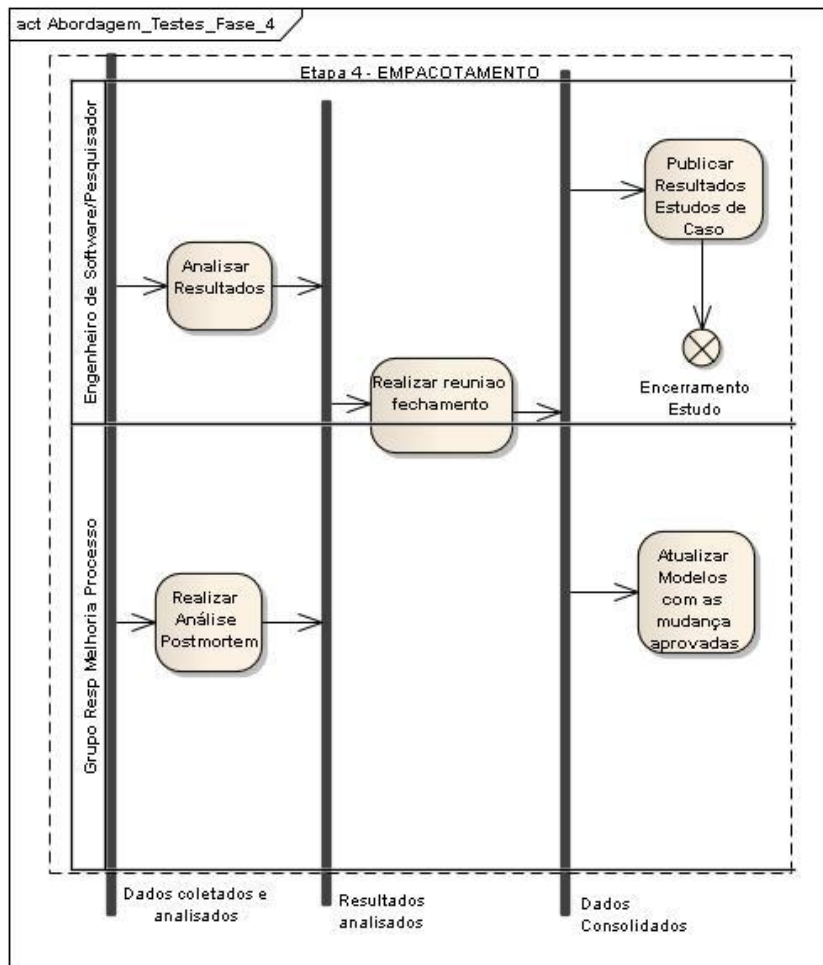


Figura 4-5 - Etapa 4 - Empacotamento

4.3 Conclusão

Neste capítulo, foram apresentados os procedimentos a serem executados em uma organização que pretende utilizar um conjunto de práticas em suas atividades de testes, com o intuito de prover agilidade na execução do processo.

Para executar o conjunto de atividades prescritas neste capítulo, o engenheiro de *software* terá a seu dispor: (a) o processo de teste baseado na norma IEEE-829, representado na especificação SPEM e publicado na Web, incluindo documentos sobre suas atividades, artefatos, perfis e diagramas de atividades; (b) uma planilha para avaliação do processo de teste e detecção de eventuais *gaps* e diferenças através da comparação das atividades de testes da organização e aquelas modeladas no processo padrão (Dias Neto, 2006); (c) O mapeamento entre as atividades de testes do processo padrão e as práticas de agilidade (Abrantes, 2012); (d) As atividades de cada etapa (caracterização, planejamento, execução e empacotamento) especificadas no SPEM e, documentadas e publicadas na Web.

A cada rodada do conjunto de atividades desta abordagem, seja por uma organização que está executando pela primeira vez ou ainda por organizações que já experimentaram sua adoção, o conjunto de práticas será selecionado de acordo com as necessidades e restrições de contexto atual, resultando, assim, em propostas de mudanças distintas e, conseqüentemente, a provisão de material de suporte específico, bem como a formulação de um estudo de caso para observação dos resultados obtidos pela inserção das práticas selecionadas na rodada corrente.

As ações propostas demandam a disponibilidade de pessoas para participar de reuniões e avaliações das mudanças que surgirão no decorrer dos trabalhos. Neste sentido, a existência na organização de um grupo de profissionais responsáveis pelos processos de engenharia de *software* poderá facilitar a aplicação dos procedimentos.

No próximo capítulo, está descrita a instanciação deste modelo em uma organização selecionada para participar do estudo o qual duas práticas ágeis foram selecionadas a partir do mapeamento entre práticas ágeis e atividades de testes.

5 Inserção de Práticas Ágeis em um Processo de Teste

Neste capítulo, foi feito um relato da execução dos procedimentos, incluindo o planejamento e execução de um estudo de caso para avaliação da adoção de práticas ágeis em uma organização, com o objetivo de observar o resultado das mudanças propostas no processo, e o uso de um editor de Casos de Testes em um projeto de software selecionado para esta finalidade.

5.1 Introdução

Briand e Labiche (2004) ao comentarem sobre as dificuldades enfrentadas pelos pesquisadores no planejamento e execução de um estudo de caso, no âmbito das atividades de teste, destacam algumas questões: (a) a pouca quantidade de dados disponíveis para realizar análise estatística (b) pouco controle sobre o treinamento de novas técnicas a serem estudadas (c) dificuldade de mitigar *confounding effects* entre os fatores estudados (uma técnica de testes, por exemplo) e outros fatores de contexto que podem influenciar nos resultados (pessoas, agentes externos dentre outros).

Dyba et al (2005) enfatizam a importância da realização de estudos de caso no contexto da indústria. Embora não seja possível nestes estudos ter um controle e rigor como na condução de experimentos controlados, eles permitem a observação *in-vivo* de como é o desempenho de determinadas técnicas atuando em determinados contextos.

Em relação aos problemas enfrentados na acumulação de evidências de forma sistemática, sinalizam que isto ocorre não somente por conta das dificuldades de agregar resultados de diferentes tipos de estudo, mas também porque estudos EBSE (*Evidence Based Software Engineering*) do mesmo fenômeno reportam resultados distintos e por vezes contraditórios. Nestes casos, não é possível integrá-las, a não ser que seja possível entender estas diferenças. Sendo assim, as informações do contexto de estudo são relevantes para a realização de avaliações futuras. Por conta disto, na etapa inicial de caracterização, será encaminhado um conjunto de ações que visam apreender o contexto da organização e suas demandas, pontos fortes e oportunidades de melhoria.

Na fase de planejamento, estão definidos os objetivos do estudo de caso, as questões de pesquisa, as métricas, hipóteses e a instrumentação necessária (o formulário de caracterização dos envolvidos, avaliação do treinamento realizado do editor, avaliação dos resultados do estudo de casos e outros formulários de apoio e registros de reuniões). Os formulários e material de suporte estão dispostos nos apêndices H, I, J e K.

Nas etapas de execução e empacotamento, o pesquisador acompanha a execução do projeto selecionado para aplicação das práticas ágeis e posteriormente realiza a coleta dos dados, documentação e análise dos resultados obtidos.

5.2 Etapa 1: Caracterização

5.2.1 Contexto da execução

Na seleção da organização na qual as propostas de inserção de agilidade seriam aplicadas, aproveitou-se a oportunidade de o pesquisador atuar como consultor técnico em uma empresa, cujas atividades de desenvolvimento de *software* estão fortemente baseadas em processos.

Possuindo um processo padrão baseado no ciclo de vida *waterfall* e outro baseado no ciclo de vida incremental, esta empresa executa um programa permanente de melhorias e adequações de seus processos acompanhadas através de um cronograma de atividades organizacionais, e conduzidas na forma de projeto com objetivos, atividades, equipe alocada e prazos estipulados. A equipe responsável pelas melhorias constituiu o grupo de processos da organização SEPG (*Software Engineering Process Group*).

Considerando os benefícios elencados por Krasner (2001) na adoção de um modelo de maturidade a fim de apoiar a evolução de processos de *software*, a organização escolheu utilizar como referência para seu programa de melhoria o modelo MPS.BR [SOFTEX, 2011] e a norma ISO 9001 [ISO/IEC, 2008b], evoluindo seus processos através da realização de três ciclos sequenciais e consecutivos, iniciando no nível G deste modelo até obter um nível de maturidade compatível com o nível C. No desenvolvimento dos produtos de *software* são utilizadas algumas das práticas de gerência de projetos descritas no método Scrum.

Por ser uma empresa de pequeno porte e com uma estrutura operacional baseada fortemente em processos, há características em suas atividades de testes que, se estudadas, poderão trazer contribuições junto aos profissionais desta área - pressionados a buscar constantemente um alto nível de qualidade a um custo mínimo.

Durante a execução deste estudo, a empresa iniciou a modelagem de processos especializados para projetos de menor porte.

Atualmente seu processo de desenvolvimento padrão possui 74 atividades dispostas em três fases: (1) Especificação e Planejamento de Projeto contendo dez macros atividades com um total de dezoito atividades (2) Análise Projeto e Construção de *Software* composto por doze macros atividades com um total de trinta e uma atividades (3) Homologação e implementação apresentando dez macros atividades em um total de vinte e cinco atividades.

As atividades de seus processos de verificação e validação foram definidas em várias etapas deste processo de desenvolvimento padrão, sendo as atividades de testes descritas juntamente com as outras atividades de desenvolvimento em uma instrução de procedimento da organização mantida em texto, formato doc.

Devido a esta disposição, na etapa de documentação das atividades do processo de testes da organização, para obter um melhor entendimento de como estas atividades foram definidas, foram documentadas no padrão SPEM todas as atividades do processo de desenvolvimento.

Além da fábrica de *software*, a organização atua como fábrica de testes em alguns projetos. Porém com um volume de demanda bem menor que quando acionada pra atuar como responsável pelo desenvolvimento do *software*.

Dentre os fatores ambientais da Empresa (como aspectos da cultura organizacional, Recursos Humanos, Infraestrutura e outros) dois fatores influenciaram nas decisões tomadas, perante as alternativas disponíveis que apareceram nesta etapa de execução:

- ✓ O tamanho da Empresa – uma organização de pequeno porte, contando em seu quadro fixo com aproximadamente 25 profissionais contratados;
- ✓ A existência de um processo padrão definido, gerenciado e definido na política organizacional da Empresa, atualmente suportado por um conjunto de processos nível C, do modelo MR-MPS-SW.

5.2.2 Documentação dos processos da organização e testes padrão.

A documentação do Processo de desenvolvimento foi produzida a partir de informações coletadas em entrevista realizada com o responsável pelo grupo SEPG e, da leitura da instrução de procedimento da organização, composta das atividades do processo padrão de desenvolvimento, os perfis dos responsáveis e participantes, artefatos requeridos e produzidos, critérios de entrada e saída, pré-atividade e pós-atividade e ferramentas.

A documentação do processo de testes utilizado como padrão, considera as informações de atividades, perfis e artefatos descritas em (DIAS NETO e TRAVASSOS, 2006).

Na etapa de projeto dos processos, foi utilizado o *software* EPF (Eclipse Process Framework) e criada uma *method library* e dois *method plug-ins*:

- ✓ Processo_Desenv_Organizacao composta das atividades, artefatos, perfis do processo de desenvolvimento e,
- ✓ Processo_Testes contendo as atividades, artefatos, perfis do processo de testes

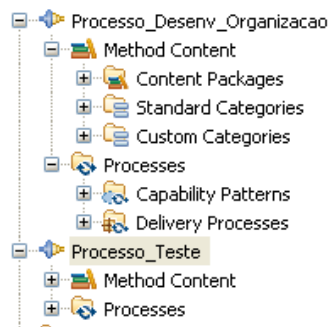


Figura 5-1 - Packages de cada Processo documentado.

E na sequência as *roles*:

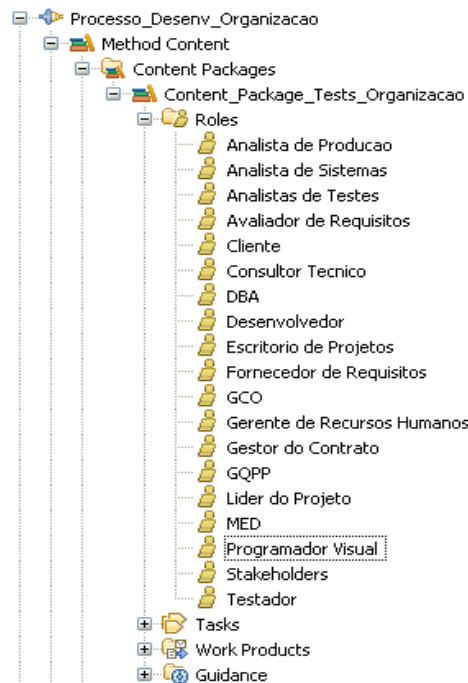


Figura 5-2 - Perfis do Processo de Desenvolvimento da Organização.

As *guidances*:

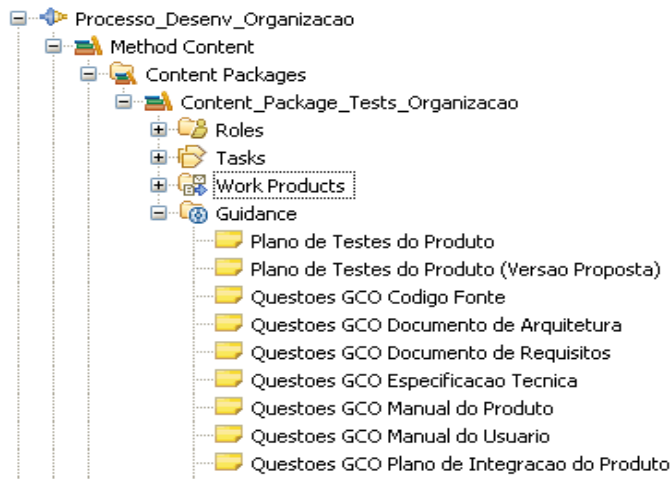


Figura 5-3 - *Templates* do Processo de Desenvolvimento da Organização.

Os *WorkProducts*:

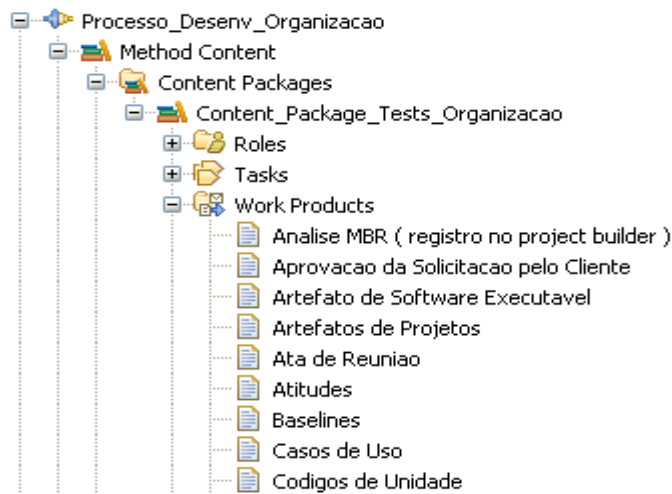


Figura 5-4 - *WorkProducts* do Processo de Desenvolvimento da Organização.

As *Tasks*:

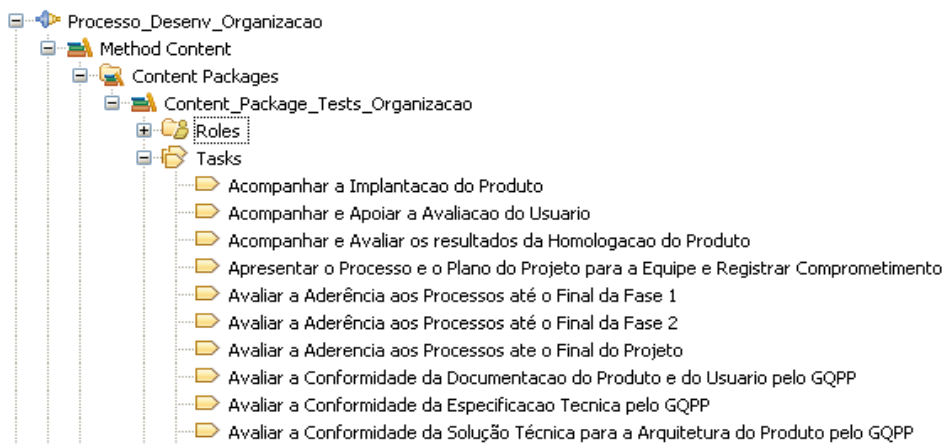


Figura 5-5 *Tasks* do Processo de Desenvolvimento de Software (PDS) da Organização

Sendo associado a cada uma das *tasks* a descrição da atividade, e seus critérios de entrada e saída:

Task: Realizar Testes do Produto

General Information
Provide general information about this task.

Name: Realizar Testes do Produto

Presentation name: Realizar Testes do Produto

Brief description: Realizar os testes do produto com base nos Casos de Teste contidos no Plano de Testes. Deve ser garantido que a implementação de cada requisito do produto seja testada para conformidade. Os resultados dos testes devem ser documentados no Relatório de Testes do Produto. Para as falhas encontradas, criar Plano de Ação (ticket). Esta atividade também identifica inconsistências entre os requisitos e planos.

Detail Information
Provide detailed information about this task.

Purpose: Realizar Testes do Produto

Main description: -Critérios de Entrada: Componentes codificados e com testes de unidade executados
-Critérios de Saída: Produto testado e Plano de Testes do Produto atualizado, caso necessário

Figura 5-6 - Descrição e Critérios de entrada/saída das *Tasks* do PDS da Organização.

Os perfis dos profissionais envolvidos:

Task: Realizar Testes do Produto

Roles
Assign the roles to perform this task.

Primary performers:

- Testador, Processo_Desenv_Organizacao/Content_Package_Tests_Organizacao

Additional performers:

- Analista de Sistemas, Processo_Desenv_Organizacao/Content_Package_Tests_Organizacao
- Analistas de Testes, Processo_Desenv_Organizacao/Content_Package_Tests_Organizacao
- Desenvolvedor, Processo_Desenv_Organizacao/Content_Package_Tests_Organizacao

Figura 5-7- Perfis das *Tasks* do PDS da Organização.

E cada *WorkProduct* utilizado:

Task: Realizar Testes do Produto

Work Products
Specify the input and output work products for this task.

Mandatory inputs:

- Artefato de Software Executavel, Processo_Desenv_Organizacao/Content_Package_Tests_Organizacao
- Plano de Testes do Produto, Processo_Desenv_Organizacao/Content_Package_Tests_Organizacao

Optional inputs:

Outputs:

- Plano de Acao, Processo_Desenv_Organizacao/Content_Package_Tests_Organizacao
- Relatorio de Testes do Produto, Processo_Desenv_Organizacao/Content_Package_Tests_Organizacao

Figura 5-8 - *WorkProducts* das *Tasks* do PDS da Organização.

As etapas do processo de desenvolvimento de *software* da organização (Especificação e Planejamento, Análise Projeto, Construção de *Software* e Homologação e Implantação) foram modeladas como *Capability Pattern*.

Presentation Name	In...	Predecessors	Model Info
Analise_Projeto_e_Construcao_do_Software	0		
Analise Projeto e Construcao do Software	1		
Refinar o Planejamento para Realizacao da Fase	2		
Monitoracao do Projeto ao Longo da Fase	9	2	
Gerencia de Requisitos ao Longo da Fase	12	9	
Realizar a Especificacao Tecnica	19	12	
Preparar Integracao do Produto	34	19	
Planejar Testes	41	34	
Construcao do Produto	48	41	
Integrar Produto	53	48	
Testes do Produto	56	53	
Encerramento da Fase	61	56	
Auditoria da Configuracao	70	61	
Aderencia dos Processos ao Final da Fase	73	70	

Figura 5-9 - *Capability Patterns* utilizados no PDS da Organização.

Posteriormente, estes blocos foram agrupados para compor a definição do processo. Na figura abaixo são exibidas as macro atividades do PDS da organização.

- Processo_Padrao_Desenvolvimento_Software_Organizacao
 - Especificacao_e_Planejamento
 - Especificacao e Planejamento do Projeto
 - Planejamento do Processo para o Projeto
 - Planejamento do Projeto
 - Monitoracao do Projeto ao longo da Fase
 - Elicitacao de Requisitos
 - Analise dos Riscos do Produto
 - Elaborar o Desenho da Arquitetura do Produto
 - Avaliar e Aprovar Requisitos e Arquitetura de Alto Nivel
 - Encerramento de Fase
 - Auditoria da Configuracao
 - Aderencia dos Processos ao Final da Fase
 - Analise_Projeto_e_Construcao_do_Software
 - Analise Projeto e Construcao do Software
 - Refinar o Planejamento para Realizacao da Fase
 - Monitoracao do Projeto ao Longo da Fase
 - Gerencia de Requisitos ao Longo da Fase
 - Realizar a Especificacao Tecnica
 - Preparar Integracao do Produto
 - Planejar Testes
 - Construcao do Produto
 - Integrar Produto
 - Testes do Produto
 - Encerramento da Fase
 - Auditoria da Configuracao
 - Aderencia dos Processos ao Final da Fase
 - Homologacao e Implantacao
 - Homologacao e Implantacao
 - Refinar o Planejamento para Realizacao de Fase
 - Monitoracao do Projeto ao Longo da Fase
 - Gerencia de requisitos ao Longo da Fase
 - Homologacao
 - Elaborar Documentacao
 - Implantacao do Produto

Figura 5-10 - Atividades do Processo de Desenvolvimento de Software da Organização

Após a modelagem realizada no EPF, os processos foram publicados em padrão *Web*, possibilitando sua visualização a partir de qualquer *browser* em ambiente de rede. Esta documentação foi entregue para a organização e publicada em sua *intranet*, e posteriormente nos endereços <http://lens.cos.ufrj.br/it73/>, <http://lens.cos.ufrj.br/it73sc/> e <http://lens.cos.ufrj.br/ieee829/>, disponível durante a realização da pesquisa. Sendo na segunda *url* (it73sc), apresentada uma especialização do processo padrão de desenvolvimento, com ciclo de vida incremental, contemplando algumas atividades da metodologia Scrum.

O conjunto das atividades que compõem cada um dos processos pode ser visualizado através de diagramas de atividades. Os diagramas das atividades de teste e desenvolvimento da organização foram construídos a partir das entrevistas com o grupo SEPG. E os diagramas de atividades de testes do processo utilizado como padrão foram desenvolvidos de acordo com a documentação do processo proposto por Dias Neto (2006). Cada diagrama permite a visualização de suas atividades e tarefas em esquema de *drill-down*.

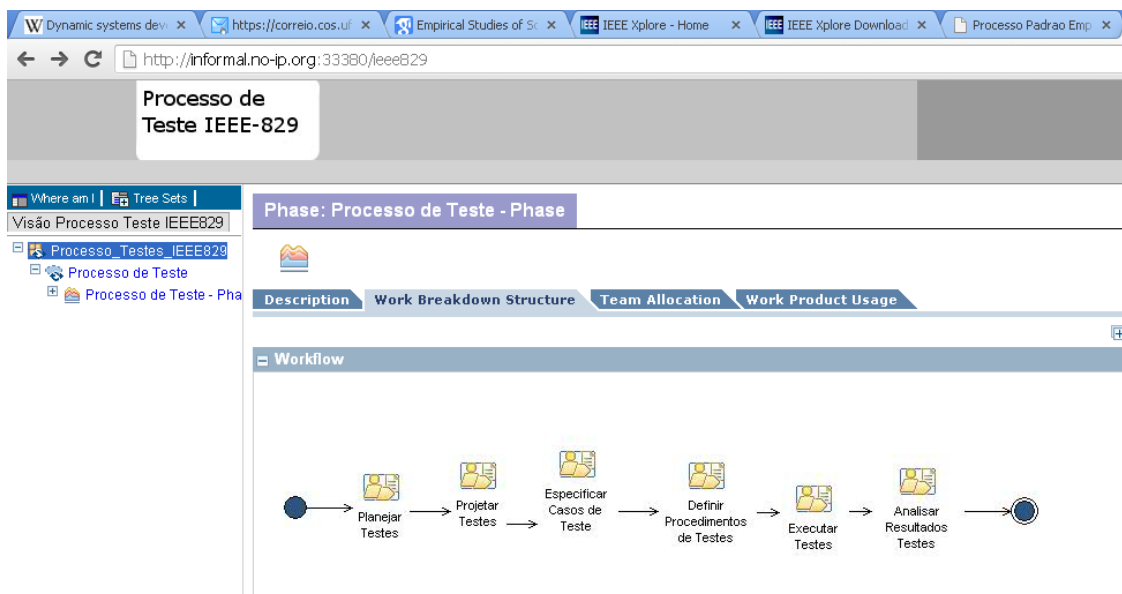


Figura 5-11 - Macro Atividades do Processo de Testes (DIAS NETO e TRAVASSOS, 2006).

Com esta documentação finalizada, iniciou-se a etapa de identificação das diferenças e características específicas de cada processo, sendo que para o processo da organização foi dada ênfase às atividades de testes e verificação.

5.2.3 Identificação das diferenças entre os componentes de cada processo

Após a etapa de documentação, foi utilizada planilha na qual foram dispostos lado-a-lado as atividades do processo de testes padrão e as atividades de testes da organização, extraídas do processo padrão de desenvolvimento.

Na primeira etapa de avaliação, em 05/08/2011, a planilha foi entregue ao Grupo de Processos da Organização e avaliados se os pontos assinalados em cada item refletiam as diferenças entre os processos.

As diferenças descritas abaixo estão agrupadas por macro atividade do Processo de Testes utilizado como padrão.

➤ **Macro Atividade Planejar Testes.**

Ao realizar a comparação entre as atividades de planejamento dos testes e os processos, verificou-se que o processo da empresa não possuía uma atividade destinada à caracterização de testes na fase 1 de planejamento e especificação do processo padrão de desenvolvimento.

A definição de uma atividade específica para a identificação dos requisitos a serem testados, e o critério de cobertura adequado para as necessidades do projeto podem dotar o processo de testes da característica ágil de convergência, na qual os riscos são atacados pró-ativamente.

Na reunião com a equipe SEPG, foi avaliado que a caracterização deveria ser incluída após a macro atividade “Avaliar e Aprovar os Requisitos e Arquitetura de Alto Nível” pois, antes disso, o que o líder e a equipe sabem sobre os requisitos é apenas o que consta no documento de termo de abertura do projeto. Se feito antes poderia limitar a efetividade da atividade de caracterização dos testes.

Também foi definido que no Plano de Riscos deveriam ser registrados os riscos associados a testes - alguns deles comuns às atividades de desenvolvimento como: equipe indisponível e cliente insatisfeito.

➤ **Especificar Casos de Teste**

O SEPG questionou a necessidade de existir atividades específicas para identificação de Casos de Testes e Procedimentos e Priorização de Procedimentos e Casos de Testes, argumentando que apesar de não estar explícito, em todo Caso de Testes definido o analista de Teste especifica a entrada, os passos e o resultado esperado.

Após esclarecimentos sobre a importância desta atividade, desde que sua definição incentiva à utilização das melhores práticas de projeto, apoiando a separação entre dados (entrada e saída) e passos a serem executados, foi combinado

que seria incluída na relação de atividades do projeto piloto selecionado para a avaliação das mudanças propostas.

➤ Analisar Resultados dos testes

Inicialmente, o pesquisador registrou a falta de uma atividade no processo da organização que possuía correspondência com a atividade “Resumir Testes” do processo padrão de Testes.

Entretanto, após reunião com equipe SEPG revisou-se este item, e constatado que na atividade “Registrar a homologação interna do produto” são avaliados os resultados da inspeção e dos testes de unidade, integração e do produto e, se necessário, é definido um Plano de Ação para correção dos eventuais problemas encontrados.

Nesta atividade, a organização aplica um *checklist* para avaliar o resultado dos testes realizados. As questões visam verificar se os resultados registrados estão de acordo com os objetivos definidos no plano de testes, em termos de acurácia e consistência.

5.2.4 Avaliação das diferenças encontradas para seleção de modificações nas atividades de teste

Após a avaliação do SEPG, das diferenças apontadas entre os processos, as possibilidades de mudanças no processo corrente foram extraídas das planilhas e dispostas em tabelas: Na primeira tabela (tabela 5-1) estão as atividades definidas no processo de testes utilizado como padrão, e que não existem no processo da organização. Na segunda tabela (tabela 5-2) estão as atividades modeladas no processo de desenvolvimento da organização e que não são contempladas no processo de testes utilizado na comparação.

Em relação ao conjunto de sugestão à respeito das atividades que não existiam no processo de desenvolvimento e testes da organização - mas que fazem parte do processo de testes padrão – o grupo SEPG decidiu em incorporar no projeto piloto as atividades de caracterização dos testes (item 1 da tabela 5-1) e a macro atividade Projetar Testes (item 2 da tabela 5-1).

Tabela 5-1 - Atividades existentes no Processo de Testes Padrão.

	Macro-Atividade Processo Padrão	Atividade(s) Processo Padrão	Possibilidade Melhoria Processo Empresa	Observações
1	(-)Planejar Testes	"Caracterizar Testes" e "Planejar Estratégia para Teste"	Estas atividades não estão definidas para serem executadas no início de cada projeto no Processo da empresa. (Atualmente é usada uma média a partir do repositório de estimativas da empresa.)	•Pode apoiar a antecipação da determinação de riscos associados aos Testes. •Pode subsidiar uma melhor determinação do esforço necessário para teste. [Convergência] Possibilidade de Melhoria Incluir atividades no processo padrão da Empresa.
2	(-)Projetar Testes	"Identificar Casos e Procedimentos de Testes"	Esta atividade não está definida no processo da Empresa. As macros "Especificar Casos de Teste" e "Definir Procedimentos de Teste" são realizadas em uma única atividade.	Pode apoiar a padronização de Projeto de Caso de Teste. [Modularidade] Possibilidade de Melhoria Incluir atividades no processo padrão da Empresa.
3	(-)Projetar Testes	Instrução de procedimento do Processo Padrão adota abordagem bottom-up (unidade, integração, software) no planejamento dos Testes.	Empresa executa as atividades de análise elicitação de requisitos em estratégia top-down porém planejamento dos testes está descrito como bottom-up.	Adequação da documentação pode fornecer maior visibilidade da estratégia de Teste [Transparência] Possibilidade de Melhoria Alterar a ordem de planejamento considerando estratégia formulada no modelo V.

Em relação às atividades modeladas no processo de desenvolvimento da organização - mas que não fazem parte do processo de testes padrão - ao examinar as questões utilizadas nas atividades de avaliação da conformidade dos Planos de Testes e Integração observa-se a proposição de questões muito abertas, e que poderiam ser complementadas por um processo de revisão por pares, trazendo mais efetividade ao processo.

Como existe uma iniciativa da organização em curso, em definir novos processos especializados levando em conta tamanho de projeto, criticidade e outros fatores, estas sugestões foram postergadas para serem utilizadas na fase de implementação destes processos.

Tabela 5-2 - Atividades existentes no Processo de Testes da Organização.

Atividade(s) Processo Empresa	Descrição	Atividade do Processo Padrão	Observações
4 (+) Avaliar Conformidade Plano de Integração e Plano Testes do Produto	Avaliar o Plano de Testes do Produto composto pelos Planos de testes de Unidades, Integração e Software e o Plano de Integração do Produto quanto ao seu formato, segundo critérios de qualidade pré-estabelecidos pelo GQPP.	N/A.	Questões de GQPP muito abertas. Exemplos : (a) As informações estão descritas com um nível de detalhes suficiente para o entendimento? (b) Inexiste uso de palavras imprecisas? (c) Os termos e conceitos estão claramente definidos? Possibilidade de Melhoria Rever as questões ou ainda a necessidade de aplicação delas.
5 (+) Avaliar o Plano de Testes do Produto	Avaliar o Plano de Testes do Produto, contemplando os testes de unidade, integração e do software, usando um checklist de questões definidas pelo Processo de Verificação.	N/A	Questões Genéricas em CheckList: Foram planejados testes para as condições de erro previstas? O Plano de Testes é consistente com a Especificação de Requisitos e o Modelo de Análise e Projeto? Os CTs são tem boa cobertura?

Quanto aos resultados da avaliação da comparação entre os artefatos de cada processo, das seis diferenças de artefatos entre os processos, foram priorizados os itens 1,2 e 3 relacionados a planejamento e projeto de testes:

Tabela 5-3 - Sugestões de mudanças nos artefatos de Testes da Organização.

Artefatos do Processo Padrão	Informações disponíveis	Observações
1 Plano de Teste	O Plano de Teste contém Identificador Único, Status, dt Criação, dt conclusão, Escopo/Requisitos Referencias a Ucs, sut, critério de Aprovação, características a serem testadas, níveis de Teste a serem adotados, necessidades de SW e HW, recursos Humanos p/ Teste, EAP atividades de Teste Processo Padrão. riscos e contingência.	Não existe no plano de Testes da Empresa os campos Critério de Aprovação e características a serem testadas. Possibilidade de Melhoria Qdo da execução das atividades "Caracterizar Testes" e "Planejar Estratégia para Testes" na etapa inicial do Projeto o Gerente de Teste deverá incluir "Critério de aprovação" e "Características a serem testadas"
2 Especificação de Projeto de Teste.	Além de identificador, datas de criação/término e itens e características a serem testadas o artefato contém: Abordagem e Critério de Cobertura, Relação de Casos de Testes, Relação de Procedimentos, Níveis de Teste a serem adotados, Critério de Aprovação/Rejeição do Teste.	Não existe este artefato no processo da empresa. Possibilidade de Melhoria A inclusão no processo das atividades "Especificar Casos de Teste" e "Definir Procedimentos de Teste" no processo sugerido pode apoiar o analista na definição dos procedimentos e Casos de Testes promovendo a separação entre entradas, resultados esperados e passos.
3 Especificação de Caso de Teste	O resumo do Teste contém um identificador, status, datas de criação e conclusão, sut, entradas, resultados Esperados, necessidade de ambiente do Caso de Teste, dependência com outros Casos de Teste.	Template de Caso de Teste da organização possui os mesmos campos do artefato do processo de teste utilizado como padrão, porém o campo "dados de entrada" nem sempre é utilizado. Possibilidade de Melhoria Padronização Design dos Casos de Teste

Tabela 5-4 Sugestões de mudanças nos artefatos de Testes da Organização.

	Artefatos do Processo Padrão	Informações disponíveis	Observações
4	Relatório de Incidente de Teste	Para cada incidente é gerado um relatório contendo data de cada incidente, um identificador, uma descrição, o procedimento executado, resultados esperados, resultados Obtidos, anomalias, e tentativas de repetição, e impacto (severidade).	Empresa utiliza ferramenta Testlink aonde estão registradas as ocorrências de execução dos Casos de Testes. Porém não utiliza uma classificação de severidade das falhas encontradas. Possibilidade de Melhoria Incluir classificação de severidade.
5	Log de Teste	Relatório do Histórico dos incidentes de Teste.	Empresa utiliza ferramenta de Bug-Tracker (Trac) para registro dos incidentes de Teste. Testador registra na ferramenta Testlink resultado de Teste e quando este apresenta uma inconsistência cadastra um TicketPlano de ação no Trac. acompanhado pelo GP.
6	Relatório de Resumo dos Testes	O resumo do Teste contém um identificador, os itens de software sobre Teste, as características, os artefatos produzidos, lista dos incidentes por trial, os desvios em relação ao planejamento (atrasos, mudanças), a abrangência dos Testes (o que não pode ser testado e a razão para o problema), a análise dos Critérios de aceitação dos itens de Teste (caso exista algum critério não atendido esta informação será registrada), o resumo de EAP das atividades de Teste.	Possibilidade de Melhoria Incluir no relatório de Post-Mortem um item específico de testes resumo da análise dos Critérios de Cobertura, desvios em relação ao planejado e atendimento dos critérios de aceitação.

Com o objetivo de apoiar a execução do projeto piloto, a ser utilizado na avaliação das mudanças propostas, foi definido conjuntamente com o SEPG a adaptação do processo atual para instanciação de um novo processo.

5.2.5 Seleção de práticas ágeis

Após a definição dos ajustes nas atividades de testes - a serem encaminhados na execução de um projeto piloto – iniciou-se o mapeamento das possibilidades de inserção de características de agilidade no processo constituído.

Para esta etapa foi utilizado o corpo de conhecimento proposto por Abrantes introduzido no capítulo 2 (ABRANTES, 2012). No Apêndice D estão listadas as nove práticas ágeis e, para cada uma delas, as atividades de testes associadas, e um detalhamento de como estas práticas podem trazer benefícios ao processo através de sua adoção.

Na avaliação das oportunidades de aplicação destas práticas nas atividades de testes foi considerado o contexto da organização e eventuais restrições. Neste sentido, ao analisar cada uma das práticas e sua eventual adoção constatou-se que, em alguns casos, a prática era parcialmente ou totalmente utilizada, visto que, pelo

fato da organização utilizar práticas de gerenciamento do método SCRUM, esta já tinha incorporado alguns métodos de trabalho que foram disseminados para as atividades de análise, construção e testes. Este foi o caso da prática de reuniões diárias utilizada pelas equipes SCRUM.

Na tabela 5-5 estas práticas e as atividades correlatas aparecem com um símbolo de *check* representando o atendimento a esta associação.

Em outros casos, as atividades de testes não utilizavam a prática avaliada, entretanto, por características da empresa, sua aplicação não fazia muito sentido ou era de difícil execução.

Isto aconteceu, por exemplo, em relação à prática metáfora que poderia ser utilizada para facilitar o entendimento de novos domínios, porém para a organização selecionada neste estudo não era útil de imediato, visto que os *softwares* desenvolvidos apresentavam o mesmo domínio de problema.

Em outros casos, a prática poderia ser útil em condições muito específicas. Este é o caso da possibilidade de inserção da prática equipe completa. Ao avaliar a pertinência de sua adoção, um dos membros da organização ao avaliar os requisitos não-funcionais de um determinado projeto sentiu a falta de um especialista em segurança para apoiar na definição deste tipo de testes. Entretanto, considerando as restrições de custo do projeto não se mostrou exequível a contratação de um especialista de testes de segurança para a realização do estudo. Na tabela 3-6, quando ocorreram situações desta natureza, as associações relacionadas aparecem marcadas com um círculo vazio.

E por último, apontadas as associações passíveis de serem realizadas pela organização. Estas aparecem na figura com um pequeno círculo preenchido.

Tabela 5-5 - Oportunidades de estudo de práticas ágeis nas atividades de teste.

Atividades	Reuniões Diárias	Liberações Frequentes	Metáfora	Equipe Completa	Cliente Presente	Design Simples	Visibilidade de Projeto	Backlog de Produto	Jogo de Planejamento
Planejar Testes	✓		○		○		●	●	●
Projetar Testes	✓		○	○	○	●	●		●
Especificar CTs	✓			○		●	●		
Definir Procedimentos	✓					●	●		
Executar Testes	✓	✓					●		
Analisar Resultados	✓	✓					●		
Monitorar e Controlar o Processo de Teste	✓	✓					●		

O resultado desta avaliação aparece nas próximas tabelas:

Tabela 5-6 Práticas *Backlog de Produto* e *Cliente Presente* X Atividades Teste

	Atividade	Embasamento: a prática	Observação
BACKLOG DE PRODUTO	Planejar Testes	Pode apoiar a definição dos itens a serem testados. Pode também facilitar um acompanhamento para manter o plano de testes atualizado.	<p>☐Tarefas de teste fazem parte do Selected Backlog definido para cada Sprint e gerenciadas conjuntamente com outras tarefas via TaskBoard.</p> <p>Intervenção: Incluir atividade "Priorizar Casos e Procedimentos de Teste" em Projetar Testes p/ auxiliar planejamento das atividades de teste</p>
	Planejar Testes	Pode auxiliar na solução de eventuais questões incidentes, bem como na priorização dos itens a serem testados e no estabelecimento de critérios p/ aceitação.	<p>☐Cliente Final não está presente o tempo todo porém participa das reuniões de planejamento do SCRUM: (a)Sprint Planning 1 (b) Sprint Planning 2 (c)Estimation (d)Review(e)Retrospective. Equipe da Fábrica de Testes não vislumbra ganhos na presença constante do cliente.</p> <p>Intervenção: Colocar o cliente fisicamente próximo aos desenvolvedores ou mover os desenvolvedores para próximo do cliente.</p>
CLIENTE PRESENTE	Projetar Testes	Pode apoiar a identificação de casos de teste.	

Tabela 5-7 Prática *Design Simples* X Atividades Teste

	Atividade	Embasamento: a prática	Observação
DESIGN SIMPLES	Projetar Testes	Pode facilitar a identificação de casos e procedimentos de teste	<p>☐Design dos Casos de Teste não segue sequencia de atividades Processo Padrão: quando as entradas e resultados esperados são levantados e na atividade seguinte os procedimentos .</p> <p>Como consequencia dados de entrada e pré-condições são codificadas nos CTs junto com os procedimentos dificultando o reuso dos Casos de Teste na etapa de projeto de novas releases - apesar dos Casos de Testes estarem rastreados a requisitos.</p>
	Especificar Casos de Teste	Pode facilitar a identificação de restrições e dependências com outros casos de teste	
	Definir procedimentos de Teste	Pode facilitar a identificação dos passos a serem seguidos durante ostestes	<p>Intervenção: Definição de um padrão que apoie a identificação dos Casos e Procedimentos.</p>

Tabela 5-8 - Atividades de Testes X Equipe Completa e Jogos de Planejamento.

	Atividade	Embasamento: a prática	Observação
EQUIPE COMPLETA	Projetar Testes	Por exemplo um especialista em segurança poderá auxiliar na identificação de casos de teste envolvendo questões de autenticação, autorização e auditoria.	<input type="checkbox"/> Falta de profissionais especializados em segurança e/ou performance por vezes é sentida. Intervenção: Agregar à equipe um profissional c/ perfil.
	Especificar Casos de Teste	Por exemplo um especialista em segurança poderá auxiliar na especificação detalhada de casos de teste envolvendo questões de autenticação, autorização e auditoria.	
JOGO DE PLANEJAMENTO	Planejar Testes	sendo contínuo e progressivo, com prioridades estabelecidas pelo cliente, pode apoiar o estabelecimento de um plano de testes alinhado com as necessidades do projeto.	<input type="checkbox"/> As histórias são escolhidas através do valor para negócio e o custo associado (incluindo o esforço necessário para a realização de testes) com a participação do cliente final, o GP e equipe do Projeto no Sprint Planning 1. Todavia, a interação entre a equipe de Testes e a equipe de desenvolvimento não se utiliza desta prática. Intervenção: Estender esta prática para a priorização das atividades de Testes entre equipe de Testes e desenvolvedores.
	Projetar Testes	com prioridades estabelecidas pelo cliente, pode apoiar o estabelecimento de prioridades no projeto de teste	

Tabela 5-9 - Atividades de Testes X Metáfora e Liberações Frequentes.

	Atividade	Embasamento: a prática	Observação
METÁFORA	Planejar Testes	apoiar a busca de um planejamento adequado para os testes.	Pode ser utilizada para fazer a ligação de um domínio conhecido com um domínio com o qual a equipe de teste não está familiarizada. Intervenção: Empresa trabalha com domínios bem definidos e estáveis.
	Projetar Testes	facilitar o projeto de teste, na identificação de casos e procedimentos de teste	
LIBERAÇÕES FREQUENTES	Executar Testes	influenciam as iterações ou quantidade de vezes que execuções de teste devem acontecer.	<input type="checkbox"/> Planejamento de Testes é realizado para todo o Sprint, sendo programada apenas uma bateria de Testes. Intervenção: Os procedimentos definidos em cada planejamento poderão ser executados em mais de uma bateria, gerando mais de um incremento, diminuindo o tempo para iniciar o teste.
	Analisar Resultados	influenciam as iterações ou quantidade de vezes que análise de resultados devem acontecer.	
	Monitorar e Controlar o Processo de Teste	influenciam as iterações ou quantidade de registros das tarefas executadas e dos resultados obtidos.	

Tabela 5-10 - Atividades de Testes X Visibilidade de Projeto e Reuniões Diárias.

	Atividade	Embasamento: a prática	Observação
VISIBILIDADE DE PROJETO	Planejar Testes	Melhorar a comunicação e auxiliar a elaboração de um plano de teste alinhado a realidade do projeto.	Demanda de maior facilidade de recuperação e reuso de Casos de Testes entre versões do software. Apesar de Casos de Testes estarem rastreados a requisitos por vezes falta diretriz para decisão/escolha de quais Casos de Testes devem ser executados em Regressão.
	Projetar Testes/ Especificar Casos de Testes / Definir Procedimentos de Teste / Executar Testes / Analisar Resultados / Monitorar e Controlar o Processo de Testes	Facilitar a integração das atividades de testes com as de desenvolvimento	Intervenção: Implementação de novas formas de visualização dos Casos de teste.
REUNIÕES DIÁRIAS	Executar Testes	melhorar a comunicação e auxiliar a elaboração de um plano de teste alinhado com a realidade do projeto.	☐ Nas reuniões diárias, testadores registram finalização de testes e liberam os Casos de Testes para os testadores. Desenvolvedores retornam com correções de bugs , facilitando a interação entre as equipes. Intervenção:
	Projetar Testes/Especificação Casos de Teste/Definir Procedimentos de Teste/Executar Testes/Analisar Resultados/Monitorar e Controlar o Processo de Teste	facilitar a integração das atividades de teste com as de desenvolvimento.	N/A. Processo atual apresenta esta característica.

5.2.5.1 Entrevistas com equipe de Testes

Em conjunto com esta avaliação foram examinados os artefatos de testes gerados pelo processo corrente, a saber: Plano de Testes, Casos de Testes desenvolvidos e Relatórios de Testes realizados.

A análise destes artefatos foi executada para dois grandes projetos mantidos pela organização: (a) Ferramenta para Instalação de Ferramenta de BI contendo 2592 Casos de Testes desenvolvidos (volume acumulado desde janeiro de 2012). (b) Ferramenta para Gerencia e Acompanhamento de Projetos com 5330 casos de Testes desenvolvidos, volume acumulado desde março de 2008. Sendo que no segundo projeto são mantidas versões distintas desta ferramenta para alguns clientes, e, por vezes, com manutenção em paralelo – em um esquema próximo a uma linha de produto de *software*.

Na entrevista com os analistas observou-se que:

- Os requisitos e Casos de Uso são cadastrados na ferramenta *Enterprise Architect* (EA, 2013). O analista de testes utiliza as descrições da mudança (documento em formato Word) e dos casos de uso para especificar os Casos de Teste. Quando ocorre manutenção, os requisitos testados apresentam, frequentemente, um nível de detalhamento que permite ao testador iniciar o desenvolvimento do Caso de Teste, antes que o analista de Negócios finalize a

especificação das mudanças no *EA*. Nos casos em que o desenvolvimento ocorre por conta da implementação de um novo módulo, ou de um conjunto de novas funcionalidades o analista de testes, na maioria das vezes começa o projeto dos Casos de Teste, somente após elaborar a especificação, o projeto físico de banco de dados foi desenvolvido, e o *wire-frame* projetado;

- Os testes mais utilizados são os testes funcionais (testes de integração que cuidam mais da preparação do ambiente, a instalação das DLL, configuração do ambiente etc.). A Técnica utilizada é caixa-preta;
- A atividade "Planejar recursos Físicos" da macro atividade "Planejar Testes" foi realizada uma única vez, no início do desenvolvimento do primeiro *release*;
- Não existe classificação de severidade de defeitos, e, conseqüentemente, não é estabelecida nas fases iniciais do projeto uma priorização para a execução dos Casos de Testes. Se porventura o tempo é insuficiente, o gerente de projeto prioriza, e se necessário negocia com o cliente uma reprogramação da implementação de funções para um próximo *Sprint*. Considerando o empenho pessoal e comprometimento de toda a equipe - observado durante a realização do estudo – percebeu-se que estes atrasos ocorreram poucas vezes;
- Quando é revelada uma falha na execução dos testes: (a) O Testador registra na ferramenta *TESTLINK*, (b) Abre um *ticket* para um desenvolvedor na ferramenta de *BUGTRACKER*. (c) O Desenvolvedor recebe a solicitação por *email*, verifica a causa da falha e confirma que a causa-raiz do problema é um defeito no *software*, e logo se inicia a correção. Ao terminar, o *ticket* retorna ao Testador. (d) O Testador avalia novamente a função. Estando tudo *Ok*, o *ticket* é fechado e registrado no *TESTLINK*. Se ainda for revelada uma falha na execução, o testador reenvia o *ticket* para o desenvolvedor;
- A opção pela execução dos testes de regressão é definida pelo analista de testes não sendo utilizada nenhuma técnica de cobertura;
- Não são desenvolvidos testes automatizados;
- Na definição dos valores de entradas não são adotadas técnicas de testes (como análise de valor-limite, classe de equivalência ou grafo de causa-efeito) sendo os valores de entrada escolhidos arbitrariamente a partir da experiência do analista de teste;

- Na etapa de projeto dos Casos de Testes dos *releases* da ferramenta de acompanhamento e gestão de projeto, os analistas de testes relataram que, em alguns casos, passos escritos anteriormente em versões anteriores do *software* são novamente informados, gerando retrabalho;
- O nível de detalhamento dos procedimentos nos Casos de Testes é baixo. Por ser sempre a mesma equipe que executa os testes, com bons conhecimentos sobre o sistema, este baixo nível de detalhamento não impacta nos tempos de execução. Contudo, quando é preciso alocar um novo profissional, demanda tempo para que este se familiarize com as telas do sistema, e possa executar os testes com bom desempenho;
- Os relatórios de testes são gerados durante as rodadas de execução e são utilizados pelo Gerente do Projeto para acompanhamento das atividades de teste. Ao final do projeto são salvos em repositório de controle de versões juntamente com plano de testes;
- Com o objetivo de obter um *feedback* à respeito do nível de satisfação de qualidade dos clientes da ferramenta de gestão, foram coletados dados de suporte de *help-desk* relativos aos últimos cinco anos. Como estes dados não estão categorizados pelo tipo do chamado (dúvida, falha reportada, elogio etc.) é difícil avaliar como evoluiu a quantidade de defeitos reportados no produto em uso. Segundo um dos gestores, a maior parte esteve associada às dúvidas de utilização do *software*. Quanto à evolução da base de clientes, não houve grandes variações que justificasse a diminuição do número de chamados (o total de clientes do produto no período permaneceu estável).

Tabela 5-11 - Quantidade de chamados Help-Desk

Ano	Quantidade de Chamados
2008	204 chamados sendo 53 clientes responsáveis pelos chamados
2009	338 chamados sendo 52 clientes responsáveis pelos chamados
2010	243 chamados sendo 54 clientes responsáveis pelos chamados
2011	170 chamados sendo 57 clientes responsáveis pelos chamados
2012	142 chamados sendo 44 clientes responsáveis pelos chamados.

5.2.5.2 Adoção de Práticas *Design* Simples e Visualização de Projeto

Na etapa das entrevistas em que foram examinados os Casos de Testes (CT) desenvolvidos para cada projeto da organização, constatou-se a ausência de um

padrão no preenchimento dos itens que compõem o modelo de CT adotado pelos projetistas de teste. Em alguns CTs examinados, as entradas de dados, pré-condições, passos estão descritas lado-a-lado na seção dos passos de cada procedimento.

O que parece ser decorrente de falta de uma atividade capaz de identificar os Casos de Testes conjuntamente com os procedimentos, para que na sequencia seja possível definir as entradas, pré-condições, resultados esperados e os passos dos procedimentos.

Tabela 5-12 - Caso de Testes Ferramenta de Gestão de Projetos – Versão corrente

Nome do Caso de Teste	ID 24292: Test Case CT002 – Verificar criação do <i>form</i> da geração do <i>Dashboard</i> de Projetos
Pré-condição	-
Preparação massa de dados	-
Entradas	
Objetivo	-
Passos	Acessar o sistema como ADM (testar também com outros usuários para verificar que somente pessoas com permissão “Acessar todos os projetos” tenham acesso à geração do relatório. Ou seja, caso não tenha tal permissão, não verá nem a opção “Visões personalizadas) Cadastrar 1 projeto com 1 componente sem fim real Acessar “Biblioteca >> Visões Personalizadas”
Resultados esperados	Exibição do formulário conforme protótipo disponível no documento de especificação técnica (página <i>Dashboard</i> de Projetos)

A ausência destas práticas impede a reutilização dos passos definidos, gerando retrabalho e dificultando o acompanhamento das atividades de projeto pelo gerente do projeto de teste, o que é justamente oposto às características *leaness* e transparência (Apêndice B). Nas tabelas 5-12 e 5-13 são exibidos dois exemplos de Casos de Testes desenvolvidos pela organização para os dois produtos de *software* mencionados:

Tabela 5-13 - Caso de Testes Ferramenta Instalador BI – Versão corrente

Nome do Caso de Teste	ID 18048: Test Case CT008 – Verificar preenchimento do campo “ <i>User</i> ”
Pré-condição	-
Preparação massa de dados	-
Entradas	
Objetivo	Estar na janela de testes em questão
Passos	- Digitar um usuário não existente - Digitar qualquer caractere em “ <i>Password</i> ” - Clique em “ <i>Next</i> ”

Resultados esperados	- Mensagem de usuário inválido
----------------------	--------------------------------

Foram examinados outros Casos de Testes e para alguns deles desenvolvidas versões obedecendo a boas práticas de projeto de separação de dados e procedimentos. Estas versões são exibidas nas tabelas 5-14 e 5-15 para o Caso de Testes ID 24292:CT002. Neste caso, foram derivados dois casos de Testes para duas situações: usuário possui permissão de acesso e usuário não possui permissão de acesso. (Tabelas 5-14 e 5-15).

Tabela 5-14 - Caso de Testes CT002A – Versão proposta

Nome do Caso de Teste	ID 24292: Test Case CT002A – Verificar criação do <i>form</i> da geração do <i>Dashboard</i> de Projetos
Pré-condição	a) Usuário estar autenticado no sistema com <i>login</i> com permissão "Acessar todos os projetos" b) Projeto cadastrado com 1 componente sem fim real
Preparação massa de dados	a) Criar usuário " <i>usuallProj</i> " e senha " <i>usuallproj</i> " com permissão de "Acessar todos os projetos". b) Criar projeto "Proj" com 1 componente sem fim real.
Entradas	Projeto X="Proj" , <i>Login</i> Y="usuallProj" , Senha S="usuallProj"
Objetivo	Visualizar "Visões Personalizadas"
Passos	1- Autenticar <i>login</i> Y senha S. 2- Selecionar o projeto X. 3- Abrir menu "Biblioteca" 4- Acessar a opção "Visões Personalizadas"
Resultados esperados	Formulário apresentado com os valores "T" e "R".

Tabela 5-15 - Caso de Testes CT002B – Versão proposta.

Nome do Caso de Teste	ID 24293: Test Case CT002B – Verificar criação do <i>form</i> da geração do <i>Dashboard</i> de Projetos
Pré-condição	a) Usuário estar autenticado no sistema com <i>login</i> que NÃO possui permissão "Acessar todos os projetos" b) Projeto cadastrado com 1 componente sem fim real
Preparação massa de dados	a) Criar usuário sem permissão de "Acessar todos os projetos". b) Criar projeto com 1 comp sem fim real.
Entradas	Projeto X = "Proj" , <i>Login</i> Y= "usunotallProj" , Senha = "usunotallProj".
Objetivo	Visualizar "Visões Personalizadas"
Passos	1- Autenticar <i>login</i> Y senha S 2- Selecionar o projeto criado X. 3- Abrir menu "Biblioteca". 4- Acessar a opção "Visões Personalizadas".
Resultados esperados	Opção "Visões Personalizadas" não estará disponível.

Como consequência desta abordagem, as responsabilidades de cada elemento do Caso de Testes ficam bem definidas: entrada de dados, resultados de saída esperados, preparação de massa de dados, verificação de pré-condição, e passos. Sendo que alguns deles poderão ser reutilizados em Casos de Testes distintos.

Esta proposta foi encaminhada em reunião, juntamente com as mudanças selecionadas, sendo combinada a execução de um estudo de caso em data a ser definida junto com a organização. (Apêndice F – Ata reunião SEPG)

Nesta etapa da pesquisa, ficou configurada a possibilidade de inserir as práticas “*Design Simples*” e “*Visibilidade de Projeto*”, relacionadas às atividades de “Projetar Testes”, “Especificar Casos de Teste” e “Definir Procedimentos de Teste” (Tabelas 5-7 e 5-10) propondo para isto um *template* baseado na norma IEEE-829, posteriormente adotado no projeto de CTs e Procedimentos, com as mudanças selecionadas na comparação entre as atividades de testes do processo de desenvolvimento e do processo de teste utilizado como padrão.

Quanto às práticas “*Backlog de Produto*” e “*Jogos de Planejamento*”, a organização achou interessante utiliza-las em um projeto futuro da fábrica de testes nas atividades de priorização de funcionalidades a serem testadas e no acompanhamento das entregas realizadas ao cliente.

Sendo assim, as práticas “*Design Simples*” e “*Visibilidade de Projeto*” foram selecionadas pelo SEPG a fim de aplicar em suas atividades de testes nos ciclos de desenvolvimento da ferramenta de gerencia de projetos.=

5.3 Etapa 2: Planejamento

5.3.1 Documentação das Mudanças Propostas

Para documentar as mudanças, gerou-se, a partir do EPF, uma nova versão do processo de teste (*Delivery Process*) envolvendo as mudanças assinaladas, para que pudessem ser acompanhadas na execução do processo no projeto piloto.

Foram exportadas as definições do processo cadastradas no EPF em um arquivo XML e posteriormente importadas para o *software MsProject*, realizando, assim, o *enactement* do processo. Porém, conforme registrado em ata de reunião (Apêndice F), a organização resolveu acompanhar o projeto através de ferramenta própria, e, para isto, incluiu na estrutura analítica do projeto as novas atividades propostas. Também optou pela Adoção do EPF para definição e manutenção do IT 7.3, que contém o seu processo de desenvolvimento padrão.

As mudanças selecionadas e aprovadas pelo SEPG foram registradas na *intranet* da organização, gerando duas novas versões: <http://lens-ese.cos.ufrj.br/it73pro> e <http://lens.cos.ufrj.br/it73scpro>.

Nas figuras abaixo estão registradas as mudanças sugeridas:

- Inserção macro atividade “Caracterizar Testes e Planejar Estratégia” na fase 1 de “Especificação e Planejamento do Projeto”;
- Inserção atividade “Projetar Testes” na macro atividade “Planejar Testes do Produto” na fase 2 de “Análise, Projeto e Construção do Software”.

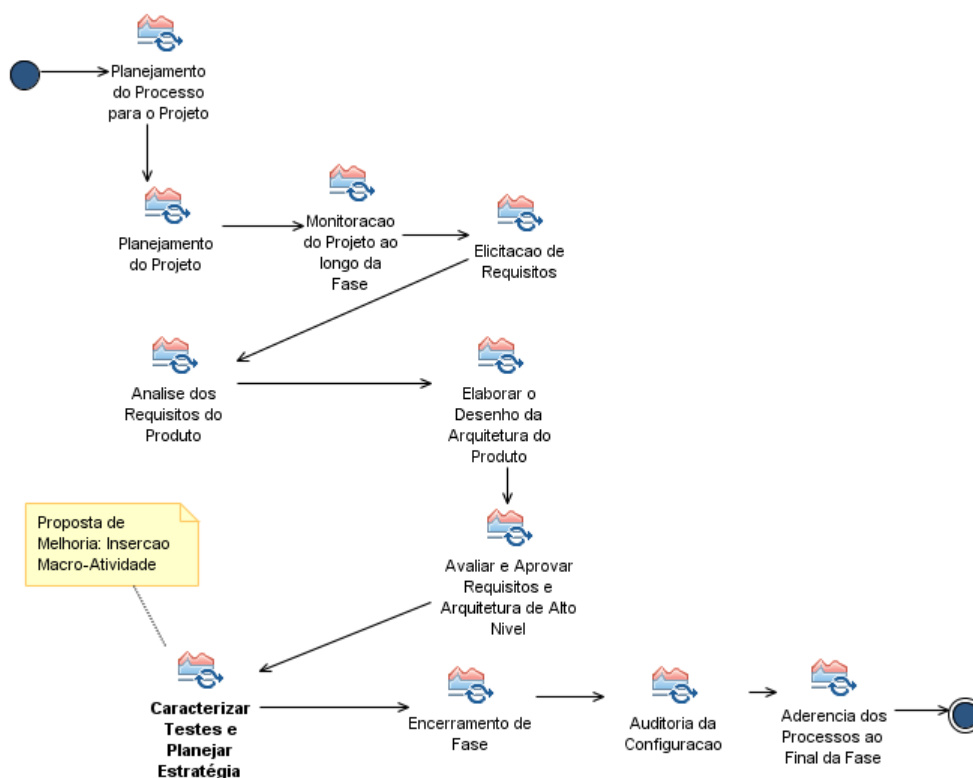


Figura 5-12 - Proposta Inclusão Macro Atividade Caracterização Testes na Fase 1.

O diagrama de atividade aparece na figura 5-13, disponível nas páginas do site publicado para acesso dos colaboradores da organização. Nela foram assinaladas (em amarelo) as mudanças sugeridas:

A atividade “Caracterizar Testes e Planejar Estratégia” é constituída de duas atividades, responsáveis respectivamente pela: caracterização dos testes e planejamento da estratégia para testar o produto:



Figura 5-13 - Atividades de Caracterização Testes e Planejamento Estratégia.

Na figura 5-15 estão documentadas as mudanças para as macro atividades da fase 2 de “Análise, Projeto e Construção do *Software*”.

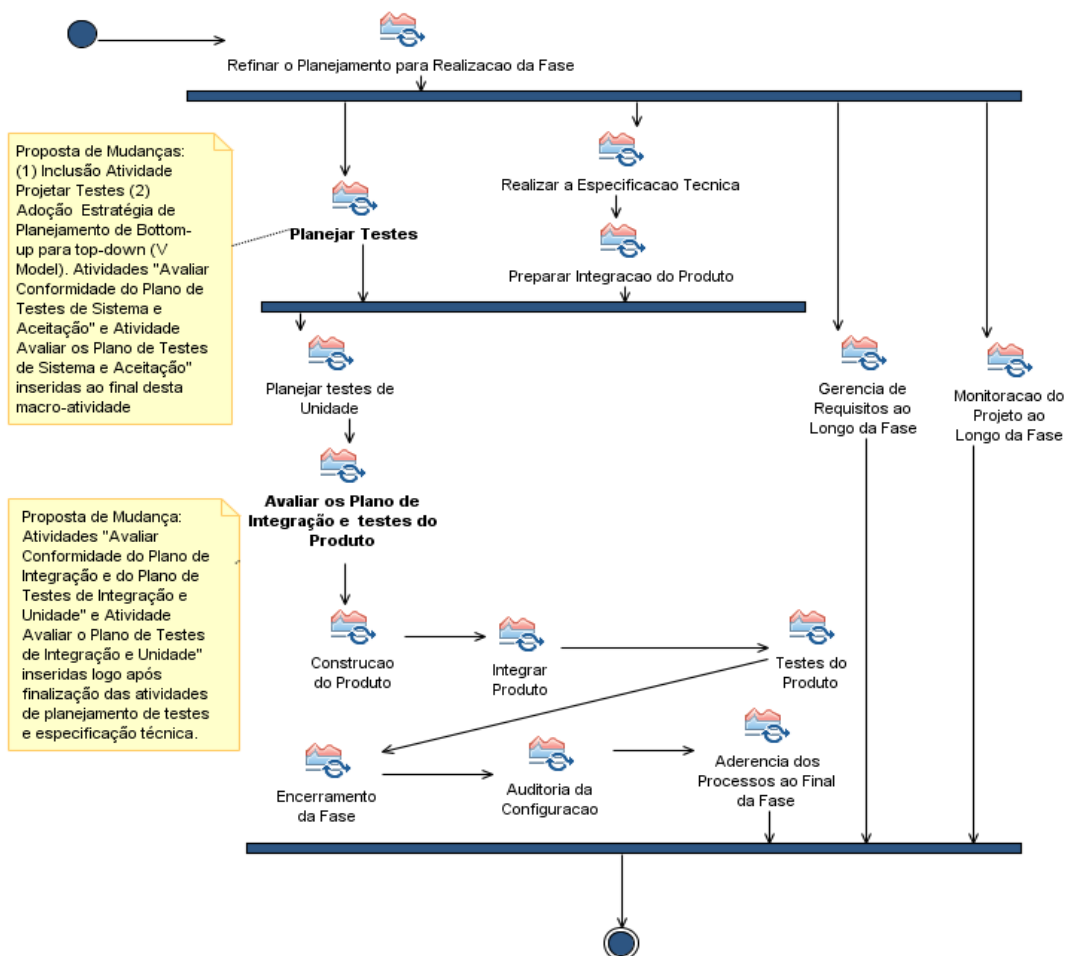


Figura 5-14 – Macro Atividades da Fase 2 Análise, Projeto e Construção de Software.

Nesta figura, estão assinaladas as atividades “Planejar Testes do Produto” e “Avaliar os Planos de Integração e testes do Produto” para contemplar a mudança sugerida na qual as atividades de planejamento de testes ocorrem paralelamente às funções “Realizar Especificação Técnica”, “Preparar Integração do Produto”.

Na figura 5-16 está o diagrama de atividades com a proposta de inclusão de uma atividade “Projetar Testes do Produto”, antecedendo a atividade já executada pela empresa denominada “Planejar os Testes do Produto”. O objetivo é assegurar a realização de uma atividade de projeto que anteceda as tarefas de desenvolvimento dos casos de testes e procedimentos, de acordo com os possíveis ganhos citados anteriormente.

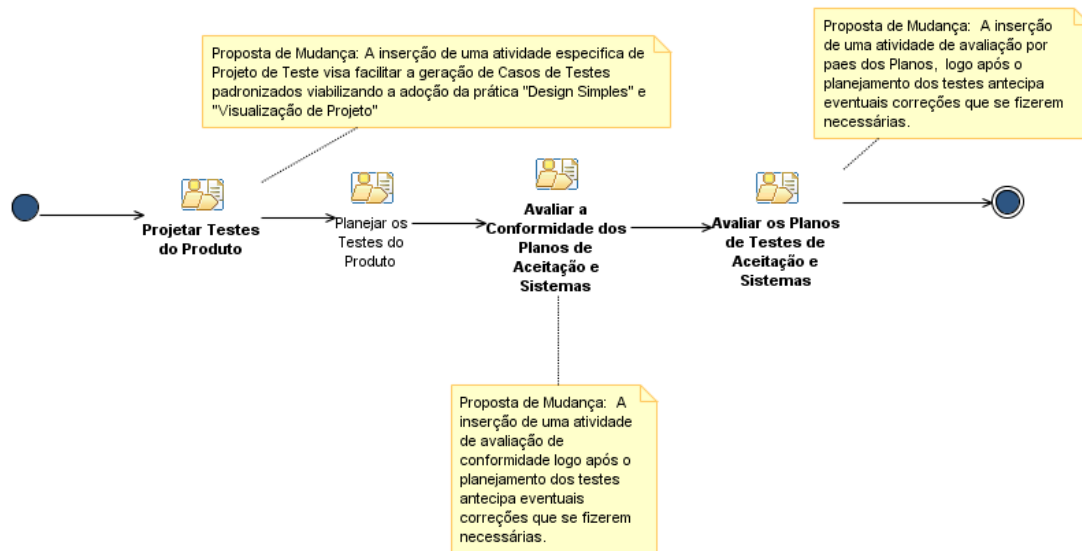


Figura 5-15 - Atividades que compõem a Macro Atividade Planejar Testes do Produto.

A atividade “Projetar Testes do Produto” é constituída por um conjunto de subatividades que visam identificar os CTs e Procedimentos, antes do início das atividades de projeto e construção destes artefatos para priorização (Figura 5-17).

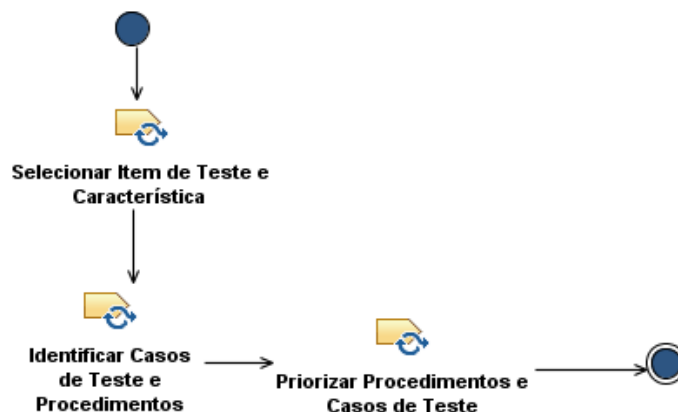


Figura 5-16 – Subatividades de “Projetar Testes”.

Com as propostas de mudanças documentadas e compartilhadas com a equipe SEPG da organização, a próxima etapa teve o objetivo de determinar como viabilizar o uso do *template* e definir um conjunto de procedimentos que apoiasse a adoção das

práticas de “Visibilidade de Projeto” e “*Design* Simples” nas atividades “Projetar Testes” “Especificar Casos de Testes” e “Definir Procedimentos de Testes”.

Quanto à proposta de adoção do modelo V cujo planejamento dos testes é executado em estratégia *top-down*, a organização adiou esta mudança para uma nova rodada dos procedimentos.

Foi realizado um levantamento inicial das métricas associadas aos processos de teste, verificação e validação. Entretanto como a organização estava iniciando um programa de especialização de processos com possibilidade de revisão destas métricas, não foram utilizados estes indicadores.

5.3.2 Fornecimento de Suporte para adoção das práticas escolhidas

Através da implementação das mudanças documentadas espera-se contemplar os seguintes tópicos relacionados à agilidade em testes:

- A inserção de uma atividade de caracterização dos testes na fase inicial do processo de desenvolvimento pode antecipar: (a) a descoberta de eventuais riscos para as atividades de teste, (b) a determinação de recursos que serão necessários e, (c) permitir que os profissionais da equipe possam iniciar um esboço do planejamento de testes de forma proativa. Esta mudança no processo poderá dotar o processo da característica de convergência, através da qual os riscos são atacados pro ativamente;
- A inserção de uma atividade de projeto de testes apoiada na adoção de um *template* visa endereçar a prática ágil “*Design* Simples” com o objetivo de facilitar a identificação dos passos dos procedimentos de Testes e apoiar a identificação de restrições na atividade “Especificar Casos de Teste”. Esta inserção de atividade no processo e a adoção deste *template*, além de apoiar o trabalho de projeto dos Casos de Testes visa também incentivar uma prática de reutilização dos ativos de teste.

Considerando a perspectiva ágil de minimizar os esforços dispendidos em trabalho desnecessário, removendo do processo de desenvolvimento toda a atividade desnecessária, a adoção de uma estratégia para eliminar a repetição rotineira de tarefas já realizadas anteriormente e facilitar a visualização dos elementos dos Casos de Testes (procedimentos, resultados esperados, pré-condições, entradas) poderá trazer agilidade às atividades de projeto e construção destes artefatos.

Contudo, para alcançar estes resultados, além das mudanças no processo, deverá ser considerada a utilização de um conjunto de diretrizes ou ainda uma ferramenta que apoie esta estratégia.

5.3.2.1 Editor de Caso de Testes (EditorCT)

Embora testes manuais sejam caracterizados por um alto nível de criatividade dos testadores, e por vezes pouco documentados, alguns princípios gerais sobre redundância de testes podem ser aplicados a este tipo de teste.

Engstrom e Runeson (2009) realizaram um estudo de caso, no contexto de uma organização, cujos produtos de uma determinada divisão apresentavam características semelhantes a uma linha de produção emergente, com o objetivo de investigar aspectos da redundância e sobreposição de Casos de Testes manuais. As questões de pesquisa visam determinar o quanto de sobreposição e execução repetida de testes ocorre, quando e porque esta sobreposição acontece e, como elementos de visualização podem apoiar a determinação de cobertura de teste, informando quais Casos de Testes fornecem cobertura aos itens de *software*.

Na etapa de projeto do estudo de caso, os autores definiram três tipos de sobreposição (*Test Overlay*): (a) Sobreposição de testes – quando uma parte do conteúdo ou todo o conteúdo de um Caso de Testes está sendo coberto em outro Caso de Testes (b) Sobreposição de *Design* de Testes – quando vários Casos de Testes estão fornecendo cobertura ao mesmo item de testes (c) Sobreposição de execução de Testes – quando são realizadas execuções distintas para atender a cobertura de um mesmo item. Uma sobreposição de execução pode ser originária de uma sobreposição de *design* de testes ou uma reexecução desnecessária de um Caso de Teste.

Associaram o termo redundância de testes às situações, em que, apesar dos Casos de Testes apresentarem diferenças consideráveis, os resultados obtidos a partir das mesmas entradas não apresentavam diferenças.

Após a realização das entrevistas, coleta e análise dos dados os resultados do estudo apontaram:

(1) A quantidade de sobreposição observada era considerável (90% de sobreposição de *design*, 75% de redundância e 96% de sobreposição na execução);

(2) A sobreposição acontecia em decorrência de um conjunto de fatores: (a) A distribuição das responsabilidades dos testes: ao dividir a responsabilidade do *design* de testes pelos membros da equipe entre plataformas, diferentes clientes e níveis de testes (unidade, integração, funcionais/sistema) aumenta a probabilidade de sobreposição. (b) inconsistência no projeto e documentação dos Casos de Testes (c)

sobreposição de execução pela falta de uma análise de necessidade de cobertura dos testes de regressão e análise das diferenças (variantes) entre produtos da mesma linha;

(3) Uma ferramenta capaz de fornecer suporte visual à gestão dos testes pode auxiliar na diminuição desta sobreposição para: (a) visualizar o propósito e o foco principal dos artefatos de testes (b) identificar itens de cobertura prioritários (c) visualizar o progresso da execução dos testes.

Apesar de o estudo ser baseado em um contexto de projeto associado a uma linha de produtos, ao descrever as ameaças de validade, os autores registram que a utilização do termo linha de produção caracteriza mais a abordagem comercial do produto desenvolvido do que enfatiza uma abordagem técnica baseada em uma SPL (*Software Product Line*).

No contexto da indústria de *software* em geral, algumas ferramentas utilizadas em larga escala por grandes equipes de testes apoiam a reutilização de ativos de testes através do reaproveitamento de um subconjunto de passos dos procedimentos de cada Caso de Testes associando-os a determinadas palavras-chaves.

Entretanto, ao examinar algumas destas ferramentas, observa-se uma abordagem que enfatiza a utilização de testes funcionais automatizados, onde os testes são gerados a partir da gravação das telas utilizadas e depois mantidos via *script* por uma equipe de testadores. O custo destas ferramentas é alto e nem sempre é levada em conta que engenheiros de testes experientes podem ser tão produtivos quanto uma abordagem baseada em ferramentas automatizadas. Após a realização de mudanças sucessivas nos requisitos, uma grande quantidade destes Casos de Testes precisa ser atualizada para poderem ser reutilizados.

Os resultados obtidos em um estudo de caso realizado na indústria, para avaliar o desempenho no uso de ferramentas automatizadas e execução de testes manuais sugerem a adoção de uma estratégia que ambas as formas de testes – manual e automatizada – sejam consideradas no planejamento e execução de testes funcionais (GRECHANIK et al, 2009).

Neste cenário, duas alternativas foram consideradas com o objetivo de mitigar a sobreposição de testes e realizar um projeto de testes que forneça transparência e visibilidade para os envolvidos: (a) o estabelecimento de *guidelines* que orientem os testadores a buscar estes princípios na execução de suas atividades (b) o uso de uma ferramenta que apoiasse o projeto e cadastramento dos Casos de Testes (CTs) funcionais, e a adoção de práticas e características de agilidade, sendo cogitado o uso de uma ferramenta *open-source* para isto. Porém a possibilidade de desenvolver os requisitos necessários para a adoção destas práticas e a oportunidade de observar

sua utilização no contexto das atividades de teste da organização incentivou a condução de um projeto de um editor de Casos de Testes.

5.3.2.1.1 Projeto do EditorCT

Uma das premissas consideradas na etapa de projeto de uma ferramenta com estas características é fornecer um conjunto de recursos, que permita o analista de teste descrever cada Caso de Testes, pois evita a criação de um projeto com complexidade desnecessária e código extra, um princípio básico da prática *Design* Simples. E a razão para propor seu desenvolvimento foi promover a adoção do *template* proposto e ao mesmo tempo reduzir o esforço associado ao cadastramento dos CTs.

Como decorrência desta abordagem, a ferramenta deve ser capaz de suportar a reutilização de blocos de procedimentos e pré-condições desenvolvidos durante a evolução dos *releases* de um determinado produto, para mitigar retrabalho na elaboração destes artefatos.

Também deverá contar com um mecanismo que facilite a busca e visualização dos blocos e Casos de Teste, de forma que os participantes de uma equipe de Testes possam compartilhar Casos de Testes desenvolvidos no decorrer da evolução de um determinado produto.

A expectativa é que, através do uso contínuo, a organização se torne apta a desenvolver uma biblioteca de componentes de procedimentos e CTs, através do cadastro de blocos e sub-rotinas recuperáveis a partir de um conjunto de palavras-chave, permitindo que o projetista utilize blocos de procedimentos cadastrados previamente.

O desenvolvimento desta ferramenta foi realizado em duas etapas:

Na primeira etapa do projeto, serão definidos os requisitos a serem considerados no desenvolvimento da ferramenta. Para apoiar esta atividade foram utilizados os resultados da revisão da literatura sobre testes e da execução do protocolo de *quasi*-revisão sistemática sobre características de agilidade em atividades de teste. Ao final desta etapa, foram gerados dois artefatos: um conjunto de requisitos funcionais e não funcionais a serem utilizados nas fases de projeto e construção da ferramenta e um diagrama contendo as entidades envolvidas e seus relacionamentos.

Na segunda etapa do projeto, foi selecionado um conjunto de funções de um projeto de *software* da Organização cujos Casos de Testes utilizarão a ferramenta. Estas funções fazem parte de um estudo de caso com o objetivo de avaliar o uso

conjunto da ferramenta em termos de flexibilidade, riqueza semântica, *acuracidade*, e as mudanças nas atividades de testes sugeridas para a organização.

Os benefícios esperados na adoção desta ferramenta são: (a) Facilitar a reutilização de procedimentos em vários CTs usando pares de entrada/saída distintas. (b) Favorecer a checagem de pré-condições estabelecidas.

Os requisitos, regras de negócio, modelos e premissas que fundamentaram a formulação e desenvolvimento do Editor de CTs estão no Apêndice K.

5.3.3 Planejamento do Estudo de Caso.

Como a organização conduz um processo de avaliação e melhoria de processo através do estabelecimento de reuniões periódicas, e planejadas em um cronograma de atividades organizacionais, a fim de obter as informações de contexto para o planejamento deste estudo, o pesquisador participou destas reuniões durante os dois primeiros meses que antecederam este planejamento, coletando informações do processo de desenvolvimento e das atividades de teste.

O foco das observações visa determinar quais as maiores dificuldades relacionado às atividades de projeto e execução de testes, do ponto de vista dos profissionais envolvidos, e como está definido o processo de testes de *software* da organização.

Após a realização destas reuniões preliminares, cada atividade e ativo do processo de testes da organização foram comparados com as atividades e produtos de trabalho de um processo padrão de teste, baseado na especificação IEEE829 – determinando quais tarefas e produtos de trabalho eram comuns entre estes processos, e quais eram específicas de cada um destes processos.

De posse desta análise, uma nova sequência de reuniões complementares foi realizada junto ao grupo SEPG - com o intuito de avaliar estas diferenças e eventualmente propor mudanças no processo de testes de *software* da organização.

Como resultado deste trabalho, as seguintes propostas de mudanças foram aprovadas: (a) Inserção de atividade para Caracterização de Testes com as atividades de análise e elicitação de requisitos na primeira fase do processo padrão da organização (b) Antecipação da atividade “Planejar Testes” para que seja realizada em paralelo com a atividade de especificação técnica. (c) Inserção de uma atividade específica para projeto de Casos de Testes através da formalização de subatividades relacionadas à identificação e priorização dos Casos de Testes, bem como a especificação dos critérios de aprovação/rejeição da característica de teste.

Depois de finalizada esta etapa, foi avaliada com os profissionais da empresa as sugestões de inserção de agilidade - mapeadas no corpo de conhecimento

proposto - quanto à sua pertinência, exequibilidade, e eventuais restrições de contexto da organização. Também foi debatido como as mudanças sugeridas no processo atual poderiam oferecer agilidade na consecução das atividades de teste.

De acordo com a experiência da equipe de testes da organização, os pontos mais críticos nas atividades de projeto e execução dizem respeito a: (a) necessidade de o testador ter um bom conhecimento e domínio do problema para executar os testes, desde que o nível de detalhamento dos Casos de Testes varia de acordo com o analista de testes responsável pela especificação, apesar de existir um padrão para projeto destes artefatos. (b) uma baixa taxa de reuso de procedimentos e casos de testes.

A partir destas observações e da separação das atividades de projeto de Casos de Testes e projeto dos procedimentos dos CTs, a oportunidade de adoção de um *template* surgiu para facilitar a especificação destes artefatos, a partir de algumas características associados à prática de agilidade “*Design Simples*” e “*Visibilidade de Projeto*”.

De acordo com a prática “*Design Simples*”, complexidade desnecessária e código extra, devem ser removidos logo que reconhecidos, não sendo apropriado incluir aspectos adicionais aos artefatos, sem uma boa justificativa para tal. A adoção desta prática visa facilitar a atividade de projeto e promover a geração de um artefato com boa inteligibilidade, facilitando a identificação de dependência e restrições entre Casos de Teste.

O objetivo é facilitar e apoiar a elaboração, e a leitura dos Casos de Testes promovendo o reuso destes artefatos. Pois é percebida pelos profissionais da equipe de Testes uma baixa taxa de reutilização dos Casos de testes entre *releases* de um mesmo produto, gerando retrabalho.

Pretende-se também, na elaboração deste *template*, facilitar a visibilidade dos procedimentos utilizados previamente, para que seja possível endereçar os requisitos derivados da prática de visibilidade de projeto. Segundo a qual, modelos e informações do projeto devem estar acessíveis a toda a equipe facilitando a identificação de Casos e Procedimentos de Teste.

Para que seja possível determinar os impactos da adoção das práticas de “*Visibilidade de Projeto*” e “*Design Simples*” no curso das atividades de projeto e construção de Casos de Teste, algumas informações serão coletadas das *baselines* dos *releases* já desenvolvidos pela organização.

Apesar de a organização possuir *baselines* com registro dos valores médios de esforço homem-hora bem como outras informações de custo e prazo, como os procedimentos associados a cada caso de testes não estão classificados por

complexidade, será realizado um trabalho prévio de categorização destes artefatos, a partir de uma taxonomia proposta. O objetivo desta tarefa é viabilizar a possibilidade de comparação entre os tempos dispendidos antes, e depois da inserção das práticas ágeis, desde que estudos de caso na indústria podem ser úteis para avaliar e monitorar o impacto da adoção de tecnologia em organizações, através da realização de medições de informações do projeto, antes e depois das mudanças sugeridas (Kitchenham, 2007).

5.3.3.1 Definição do Estudo Experimental

5.3.3.1.1 Objetivo do estudo:

Observar a inserção das práticas ágeis “Visibilidade de projeto” e “*Design* Simples” através da proposição de mudanças nas atividades de projeto de teste, e adoção de um *template* para projeto e construção de Casos de Teste, em uma organização com um processo de testes de *software* estabelecido - com respeito ao favorecimento da visualização dos procedimentos e Casos de Testes e facilidade de projeto e reuso destes artefatos.

5.3.3.1.2 Objetivo Global:

Avaliar a adoção das práticas de “Visibilidade de projeto” e “*Design* Simples” no âmbito das atividades de projeto e construção de Casos de Testes e Procedimentos na promoção de maior visualização de Casos de Testes e facilitação de reuso no projeto destes artefatos.

5.3.3.1.3 Objetivos específicos:

O objetivo deste estudo é descrito através do paradigma GQM [Basili e Rombach, 1998]:

Analisar: a adoção das práticas de “Visibilidade de projeto” e “*Design* Simples” na execução de um processo de teste.

Com o propósito: de caracterizar.

Com respeito à: capacidade de visualização e facilitação de reuso e execução das atividades de projeto e construção de Casos de Testes.

Sob o ponto de vista dos: profissionais de uma equipe de teste.

No contexto de: times de testes de *software* trabalhando em aplicações do domínio Gerenciamento de Projetos adotando um processo adaptado de testes a partir da norma IEEE-829, com uso de um *template* para procedimentos e Casos de Teste.

5.3.3.1.4 Foco da qualidade:

Visualização e Reúso de Procedimentos e Casos de Teste.

5.3.3.1.5 Contexto:

O estudo é conduzido em uma organização cujos processos de *software* foram avaliados através do modelo MPS_BR Nível C com carência de visualização e padronização de procedimentos e casos de testes, apresentando baixa taxa de reúso destes artefatos entre os *releases* de seu produto.

Preferencialmente deverá ser aplicado em *releases* de manutenção de um produto quando maximizada a oportunidade de observar reúso de procedimentos de Casos de Testes pela necessidade de manutenção destes artefatos, por conta de mudanças de requisitos funcionais.

5.3.3.1.6 Questões:

R1) Uma equipe de testes executando um conjunto de atividades responsáveis pela caracterização, identificação e projeto de procedimentos e Casos de Testes (baseado na norma IEEE-829) e adotando um *template* sugerido para a realização das atividades de projeto - com o objetivo de promover o *design* simples, visibilidade de projeto e reúso dos artefatos de testes - tende a reutilizar uma quantidade maior destes artefatos, do que equipes executando um processo no qual estas atividades não foram prescritas e executadas? (M1 e M2)

R2) O esforço dispendido nas atividades de Projeto e Construção dos Casos de Testes e procedimentos pode ser menor em equipes que adotaram o processo modificado e o *template* proposto? (M3)

R3) O esforço dispendido na execução dos Casos de Testes pode ser menor em equipes que adotaram o processo modificado e o *template* proposto? (M4)

5.3.3.1.7 Métricas:

M1) Quantidade de Casos de Testes categorizados por complexidade por *Release*.

M2) Quantidade de procedimentos de Casos de Testes categorizados por complexidade que foram reutilizados por *Release*.

M3) Esforço médio em homens-hora dispendido por cada analista de testes para realização das atividades de projeto e construção categorizados por complexidade dos Casos de Testes.

M4) Esforço médio em homens-hora das atividades de execução realizadas por cada testador para cada categoria de complexidade dos Casos de Testes.

M5) Quantidade total de passos associados a cada Caso de Teste compreendendo os passos necessários para atender as pré-condições, os procedimentos de execução realizados pelo testador e os passos relacionados à verificação dos resultados esperados.

Na categorização dos Casos de Testes será utilizada uma adaptação de uma fórmula proposta para determinação de complexidade de Caso de Uso. Esta fórmula considera o número de ações, fluxos alternativos e regras associadas a cada um destes artefatos [Massolar, 2011]. Nesta classificação, serão considerados os passos de cada procedimento de Caso de Teste, pré-condições e das rotinas de verificação dos resultados esperados.

Para que seja possível assegurar coerência na comparação entre as medidas obtidas em cada Caso de Testes é desejável que o domínio do problema seja o mesmo, e os Casos de Testes sejam projetados e construídos preferencialmente pela mesma equipe de testes que projetou os artefatos em *releases* anteriores do projeto.

Os Casos de Uso que originaram os Casos de Testes devem ser escritos pela mesma equipe de desenvolvimento – de forma a preservar o estilo de redação.

5.3.3.1.8 Confounding Factors:

- Tempo necessário para obter fluência no uso do *template* proposto.
- Características dos requisitos dos *releases* do *software* que participarão do estudo resultando em baixa probabilidade de oportunidade de reúso de CTs.
- Precisão dos registros de esforço dispendido em cada atividade de teste.
- Efetividade do treinamento sobre o uso do *template*.
- Critério para classificação da complexidade dos CTs analisados.
- A forma escolhida para a implementação das práticas de “*Design Simples*” e “*Visibilidade de Projeto*” nas atividades de projeto de Testes.

5.3.3.2 Planejamento

5.3.3.2.1 Formulação da hipótese;

R1: Uma equipe de testes executando um conjunto de atividades responsáveis pela caracterização, identificação e projeto de procedimentos e Casos de

Testes (baseado na norma IEEE-829) e que adota um *template* sugerido para a realização das atividades de projeto - com o objetivo de promover o *design* simples, visibilidade de projeto e reuso dos artefatos de testes - tende a reutilizar uma quantidade maior de procedimentos associados a Casos de Testes, do que equipes executando um processo no qual estas atividades não foram prescritas e executadas?

R1.1: Em relação a Casos de Testes de baixa complexidade:

Hipótese Nula (H₀1_1): Não há diferença significativa entre a quantidade de procedimentos de Casos de Testes de baixa complexidade reutilizados na execução de um processo de testes Pt1, para o qual foi definido um conjunto de atividades de caracterização, priorização e projeto de CTs baseado em um *template*, e a quantidade de procedimentos reutilizados em um processo de Testes Pt2 no qual estas atividades não foram prescritas e executadas.

H₀1_1: (QtdCtRB_{PT1}) = (QtdCtRB_{PT2}), onde:

- **QtdCtRB_{PT1}** = Quantidade de Casos de Testes de Baixa Complexidade Reutilizados em um processo de Testes em que foram aplicadas as mudanças propostas.
- **QtdCtRB_{PT2}** = Quantidade de Casos de Testes de Baixa Complexidade Reutilizados em um processo de Testes em que não foram aplicadas as mudanças propostas.

Hipótese Alternativa 1(H₁1): A quantidade de procedimentos reutilizados de Casos de Testes de baixa complexidade na execução de um processo de testes Pt1 cujas atividades adotaram o *template* e as mudanças sugeridas é maior do que a quantidade de procedimentos reutilizados na execução de um processo Pt2 no qual não foram adotadas as mudanças sugeridas e o uso do *template*.

H₁1_1: (QtdCtRB_{PT1}) > (QtdCtRB_{PT2})

Hipótese Alternativa (H₂1): A quantidade de procedimentos reutilizados de Casos de Testes de baixa complexidade na execução de um processo de testes Pt1 cujas atividades adotaram o *template*, e as mudanças sugeridas é menor do que a quantidade de procedimentos reutilizados na execução de um processo Pt2 no qual não foram adotadas as mudanças sugeridas e o uso do *template*.

H₂1_1: (QtdCtRB_{PT1}) < (QtdCtRB_{PT2})

R1.2: Em relação a Casos de testes de média complexidade.

Hipótese Nula (H₀1_2): Não há diferença significativa entre a quantidade de procedimentos de Casos de Testes de média complexidade reutilizados na execução do processo de testes Pt1, se comparados à quantidade de procedimentos de Casos de Testes de média complexidade do processo de Testes Pt2.

H₀1_2: (QtdCtRM_{PT1}) = (QtdCtRM_{PT2}), onde:

- **QtdCtRM_{PT1}** = Quantidade de Casos de Testes de Média Complexidade Reutilizados em um processo de Testes em que foram aplicadas as mudanças propostas.
- **QtdCtRM_{PT2}** = Quantidade de Casos de Testes de Média Complexidade Reutilizados em um processo de Testes em que não foram aplicadas as mudanças propostas.

Hipótese Alternativa 1 (H₁1_2): A quantidade de procedimentos de Casos de Testes de média complexidade reutilizados na execução do processo de testes Pt1 é maior se comparada à quantidade de procedimentos de Casos de Testes de média complexidade do processo de Testes Pt2.

H₁1_2: (QtdCtRM_{PT1}) > (QtdCtRM_{PT2}),

Hipótese Alternativa 2 (H₂1_2): A quantidade de procedimentos de Casos de Testes de média complexidade reutilizados na execução do processo de testes Pt1 é menor se comparada à quantidade de procedimentos de Casos de Testes de média complexidade do processo de Testes Pt2.

H₂1_2: (QtdCtRM_{PT1}) < (QtdCtRM_{PT2}),

R1.3: Em relação a Casos de testes de alta complexidade.

Hipótese Nula (H₀1_3): Não há diferença significativa entre a quantidade de procedimentos de Casos de Testes de alta complexidade reutilizados na execução do processo de testes Pt1, se comparados à quantidade de procedimentos de Casos de Testes de alta complexidade do processo de Testes Pt2.

H₀1_3: (QtdCtRA_{PT1}) = (QtdCtRA_{PT2}), onde:

- **QtdCtRA_{PT1}** = Quantidade de Casos de Testes de Alta Complexidade Reutilizados em um processo de Testes em que foram aplicadas as mudanças propostas.

- $QtdCtRA_{PT2}$ = Quantidade de Casos de Testes de Alta Complexidade Reutilizados em um processo de Testes em que não foram aplicadas as mudanças propostas.

Hipótese Alternativa 1 (H₁1_3): A quantidade de procedimentos de Casos de Testes de alta complexidade reutilizados na execução do processo de testes Pt1, é maior se comparada à quantidade de procedimentos de Casos de Testes de alta complexidade do processo de Testes Pt2.

$$H_{11_3}: (QtdCtRA_{PT1}) > (QtdCtRA_{PT2}),$$

Hipótese Alternativa 2 (H₂1_3): A quantidade de procedimentos de Casos de Testes de alta complexidade reutilizados na execução do processo de testes Pt1, é menor se comparada à quantidade de procedimentos de Casos de Testes de alta complexidade do processo de Testes Pt2.

$$H_{21_3}: (QtdCtRA_{PT1}) < (QtdCtRA_{PT2}),$$

R2: O esforço dispendido nas atividades de Projeto e Construção dos Casos de Testes e procedimentos é menor em equipes que adotaram o processo modificado e o *template* proposto?

R2.1: Em relação a Casos de Testes de baixa complexidade:

Hipótese Nula (H₀2_1): Não há diferença significativa nos esforços dispendidos na etapa de desenvolvimento dos procedimentos de Casos de Testes por indivíduos adotando o processo de testes Pt1 com as mudanças sugeridas e o *template* proposto, se comparados com os esforços dispendidos pelos mesmos indivíduos, desenvolvendo os procedimentos na versão do processo Pt2 anterior às mudanças propostas, em Casos de testes de baixa complexidade.

$$H_{02_1}: (EDCtB_{PT1}) = (EDCtB_{PT2}), \text{ onde:}$$

- $EDCtB_{PT1}$ = Esforço dispendido no desenvolvimento de Casos de Testes de Baixa Complexidade em um processo de Testes em que foram aplicadas as mudanças propostas.
- $EDCtB_{PT2}$ = Esforço dispendido no desenvolvimento de Casos de Testes de Baixa Complexidade em um processo de Testes em que não foram aplicadas as mudanças propostas.

Hipótese Alternativa 1(H₁2_1): Os esforços dispendidos na etapa de desenvolvimento dos Casos de Testes de baixa complexidade por

indivíduos que adotam as mudanças sugeridas no processo de testes Pt1 é menor do que os esforços dispendidos pelos mesmos indivíduos desenvolvendo os procedimentos na versão do processo Pt2.

$$H_{12_1}: (EDCtB_{PT1}) < (EDCtB_{PT2})$$

Hipótese Alternativa 2(H_{22_1}): Os esforços dispendidos na etapa de desenvolvimento dos Casos de Testes por indivíduos adotando as mudanças sugeridas no processo de testes Pt1 é maior do que os esforços dispendidos pelos mesmos indivíduos desenvolvendo os procedimentos na versão do processo Pt2.

$$H_{22_1}: (EDCtB_{PT1}) > (EDCtB_{PT2})$$

R2.2: Em relação a Casos de Testes de média complexidade:

Hipótese Nula (H_{02_2}): Não há diferença significativa nos esforços dispendidos na etapa de desenvolvimento dos procedimentos de Cts de média complexidade por indivíduos adotando o processo de testes Pt1 se comparados com os esforços dispendidos pelos mesmos indivíduos desenvolvendo os procedimentos na versão do processo Pt2.

$$H_{02_2}: (EDCtM_{PT1}) = (EDCtM_{PT2}), \text{ onde:}$$

- **EDCtM_{PT1}** = Esforço dispendido no desenvolvimento de Casos de Testes de Média Complexidade em um processo de Testes em que foram aplicadas as mudanças propostas.
- **EDCtM_{PT2}** = Esforço dispendido no desenvolvimento de Casos de Testes de Média Complexidade em um processo de Testes em que não foram aplicadas as mudanças propostas.

Hipótese Alternativa 1(H_{12_2}): Os esforços dispendidos na etapa de desenvolvimento dos Casos de Testes de média complexidade por indivíduos adotando as mudanças sugeridas no processo de testes Pt1 é menor do que os esforços dispendidos por estes mesmos indivíduos desenvolvendo os procedimentos na versão do processo Pt2.

$$H_{12_2}: (EDCtM_{PT1}) < (EDCtM_{PT2})$$

Hipótese Alternativa 2(H_{22_2}): Os esforços dispendidos na etapa de desenvolvimento dos Casos de Testes de média complexidade por indivíduos adotando as mudanças sugeridas no processo de testes Pt1 é maior do que os esforços dispendidos por outros indivíduos desenvolvendo os procedimentos na versão do processo Pt2.

$$H_{22_2}: (EDCtM_{PT1}) > (EDCtM_{PT2})$$

R3: O esforço dispendido na execução dos Casos de Testes é menor em equipes que adotaram o processo modificado e o *template* proposto?

R3.1: Em relação a Casos de Testes de baixa complexidade:

Hipótese Nula (H_{03_1}): Não há diferença significativa nos esforços dispendidos na etapa de execução dos procedimentos de Casos de Testes por indivíduos adotando o processo de testes Pt1 com as mudanças sugeridas e o *template* proposto, se comparados com os esforços dispendidos pelos mesmos indivíduos executando os procedimentos na versão do processo Pt2 anterior às mudanças propostas, em Casos de testes de baixa complexidade.

H_{03_1}: (EECtB_{PT1}) = (EECtB_{PT2}), onde:

- **EECtB_{PT1}** = Esforço dispendido na execução de Casos de Testes de Baixa Complexidade em um processo de Testes em que foram aplicadas as mudanças propostas.
- **EECtB_{PT2}** = Esforço dispendido na execução de Casos de Testes de Baixa Complexidade em um processo de Testes em que não foram aplicadas as mudanças propostas.

Hipótese Alternativa 1(H_{13_1}): Os esforços dispendidos na etapa de execução dos Casos de Testes de baixa complexidade por indivíduos adotando as mudanças sugeridas no processo de testes Pt1 é menor do que os esforços dispendidos pelos mesmos indivíduos executando os procedimentos na versão do processo Pt2.

H_{12_1}: (EECtB_{PT1}) < (EECtB_{PT2})

Hipótese Alternativa 2(H_{23_1}): Os esforços dispendidos na etapa de execução dos Casos de Testes por indivíduos adotando as mudanças sugeridas no processo de testes Pt1 é maior do que os esforços dispendidos pelos mesmos indivíduos executando os procedimentos na versão do processo Pt2.

H_{22_1}: (EECtB_{PT1}) > (EECtB_{PT2})

R3.2: Em relação a Casos de Testes de média complexidade:

Hipótese Nula (H_{03_2}): Não há diferença significativa nos esforços dispendidos na etapa de execução dos procedimentos de CTs de média complexidade por indivíduos adotando o processo de testes Pt1 se

comparados com os esforços dispendidos pelos mesmos indivíduos executando os procedimentos na versão do processo Pt2.

H_{03_2}: (EECtM_{PT1}) = (EECtM_{PT2}), onde:

- **EECtM_{PT1}** = Esforço dispendido na execução de Casos de Testes de Média Complexidade em um processo de Testes em que foram aplicadas as mudanças propostas.
- **EECtM_{PT2}** = Esforço dispendido na execução de Casos de Testes de Média Complexidade em um processo de Testes em que não foram aplicadas as mudanças propostas.

Hipótese Alternativa 1(H_{13_2}): Os esforços dispendidos na etapa de execução dos Casos de Testes de média complexidade por indivíduos adotando as mudanças sugeridas no processo de testes Pt1 é menor do que os esforços dispendidos por estes mesmos indivíduos executando os procedimentos na versão do processo Pt2.

H_{13_2}: (EECtM_{PT1}) < (EECtM_{PT2})

Hipótese Alternativa 2(H_{23_2}): Os esforços dispendidos na etapa de execução dos Casos de Testes de média complexidade por indivíduos adotando as mudanças sugeridas no processo de testes Pt1 é maior do que os esforços dispendidos por outros indivíduos executando os procedimentos na versão do processo Pt2.

H_{23_2}: (EECtM_{PT1}) > (EECtM_{PT2})

R3.3: Em relação a Casos de Testes de alta complexidade:

Hipótese Nula (H_{03_3}): Não há diferença significativa nos esforços dispendidos na etapa de execução dos procedimentos de Cts de alta complexidade por indivíduos adotando o processo de testes Pt1 se comparados com os esforços dispendidos pelos mesmos indivíduos executando os procedimentos na versão do processo Pt2.

H_{03_3}: (EECtA_{PT1}) = (EECtA_{PT2}), onde:

- **EECtA_{PT1}** = Esforço dispendido na execução de Casos de Testes de Alta Complexidade em um processo de Testes em que foram aplicadas as mudanças propostas.
- **EECtA_{PT2}** = Esforço dispendido na execução de Casos de Testes de Alta Complexidade em um processo de Testes em que não foram aplicadas as mudanças propostas.

Hipótese Alternativa 1(H_{13_3}): Os esforços dispendidos na etapa de execução dos Casos de Testes de alta complexidade por indivíduos adotando as mudanças sugeridas no processo de testes Pt1 é menor do que os esforços dispendidos por estes mesmos indivíduos executando os procedimentos na versão do processo Pt2.

$$H_{13_3}: (EECtA_{PT1}) < (EECtA_{PT2})$$

Hipótese Alternativa 2(H_{23_3}): Os esforços dispendidos na etapa de execução dos Casos de Testes de alta complexidade por indivíduos adotando as mudanças sugeridas no processo de testes Pt1 é maior do que os esforços dispendidos por outros indivíduos executando os procedimentos na versão do processo Pt2.

$$H_{23_3}: (EECtA_{PT1}) > (EECtA_{PT2})$$

5.3.3.2.2 Seleção de variáveis:

- Variáveis independentes:
 - ✓ Estratégia para Projeto e Construção de Casos de Testes;
 - ✓ Domínio da Aplicação;
 - ✓ Perfil do testador e do analista. (Deverá ser preenchido formulário de caracterização para determinação do perfil e senioridade de cada analista e testador que irá participar do estudo).
- Variáveis dependentes:
 - ✓ Quantidade de procedimentos reutilizados (classificados por complexidade);
 - ✓ Quantidade de CTs reutilizados (classificados por complexidade);
 - ✓ Esforço dispendido por complexidade de Caso de Testes para cada indivíduo participante do estudo para projetar e construir os Testes;
 - ✓ Esforço dispendido na execução dos Casos de Testes categorizados por complexidade.

5.3.3.2.3 Seleção de indivíduos:

- Será formado um grupo constituído por funcionários da organização contendo, no mínimo, um representante de cada um dos perfis abaixo:
 - ✓ Analista de Testes (Pleno) - AT
 - ✓ Testador (Junior) – TE

5.3.3.2.4 Recursos:

- Ambiente de desenvolvimento da organização selecionada para o estudo: computadores e *software* configurados com as ferramentas necessárias para a execução e registro dos resultados, e ferramenta de edição de Casos de Testes.

5.3.3.2.5 Projeto do Experimento:

- Fator: Estratégia para Projeto e Construção de Casos de Testes
 - Tratamentos:
 - Com a adoção do processo de testes contemplando as mudanças sugeridas e de um *template* baseado na norma IEEE-829 para inserção das práticas ágeis;
 - Com a adoção do processo de testes sem a implementação das mudanças sugeridas;
- Durante a execução do estudo deverá ser utilizado o mesmo domínio de aplicação em pelo menos dois ciclos de desenvolvimento;
- Será fornecida uma ferramenta para edição de CTs utilizada pela organização nas atividades de projeto e construção dos CTs cujo objetivo é facilitar a adoção do padrão e visualização destes artefatos.

Tabela 5-16 – Eventos programados pelo estudo de caso

Identificação do Evento	Atividades	Descrição	Participantes
EVENTO 1	Projeto e Construção de CTs de Sistema para Gestão de Projetos	<i>Adoção de versão de Processo de Teste antes das mudanças propostas e uso do TestLink.</i>	AT
	Execução de CTs de Sistema para Gestão de Projetos	<i>Adoção de versão de Processo de Teste antes das mudanças propostas e uso do TestLink.</i>	TE
	Coleta dos esforços realizados e Procedimentos de CTs reutilizados	<i>Recuperação de dados de baselines armazenadas nos repositórios de projetos da Organização</i>	PE
EVENTO 2	Treinamento	<i>Treinamento para uso do Template, Editor</i>	AT, TE e PE
	Avaliação do Treinamento	<i>Aplicação de formulário de avaliação para participantes</i>	AT e TE
EVENTO 3	Projeto e Construção de CTs de Sistema para Gestão de Projetos	<i>Adoção do Processo de Testes contendo mudanças sugeridas e Template baseado na IEEE-829-Primeiro Ciclo</i>	AT
	Execução de Casos de Testes de Sistema para Gestão de Projetos	<i>Adoção do Processo de testes contendo mudanças sugeridas e Template baseado na IEEE-829 – Primeiro Ciclo</i>	TE
	Coleta dos esforços realizados e Procedimentos de CTs reutilizados	-	PE

EVENTO 4	Projeto e Construção de CTs de Sistema para Gestão de Projetos	<i>Adoção do Processo de Testes contendo mudanças sugeridas e Template baseado na IEEE-829- Segundo Ciclo</i>	AT
	Execução de CTs de Sistema para Gestão de Projetos	<i>Adoção do Processo de Testes contendo mudanças sugeridas e Template baseado na IEEE-829 – Segundo Ciclo</i>	TE
	Coleta dos esforços realizados e Procedimentos de CTs reutilizados	-	PE

Legenda: AT – Analista de Testes TE – Testador PE - Pesquisador

5.3.3.2.6 Instrumentação:

- Formulário de consentimento individual para utilização dos resultados;
- Formulário para caracterização do perfil e senioridade dos profissionais envolvidos;
- Lista dos participantes;
- *Template* de Casos de Testes;
- Material de Treinamento do *template* de Casos de Testes;
- Casos de Uso (s) selecionado(s) para o estudo;
- Material de Treinamento para utilização da Ferramenta de edição de Casos de Testes;
- Questionário de avaliação do treinamento realizado pelos participantes;
- Editor para registro dos Casos de Testes e procedimentos reutilizados;
- Formulário para registro dos tempos de execução dos Casos de Testes ou uso de alguma ferramenta de controle de horas trabalhadas;
- Formulário para registro dos procedimentos e Casos de Testes reutilizados ou uso de alguma ferramenta de controle de horas trabalhadas;
- Questionário de avaliação dos participantes em relação às mudanças implementadas.

5.3.3.2.7 Mecanismo de análise:

Além de realizar a coleta das medidas e uso de estatística descritiva, será distribuído um questionário aos participantes do estudo a fim de apoiar uma avaliação qualitativa das propostas de inserção de práticas ágeis nas atividades de testes realizadas.

5.3.3.2.8 Validação dos Resultados:

- Validade interna:

Na etapa de planejamento, foram elicitados os seguintes fatores de risco de validade interna: (a) tempo para obter fluência no uso do *template* de Casos

de Testes, (b) características dos requisitos dos *releases* do *software* que participarão do estudo que resulta em uma quantidade inexpressiva de procedimentos a serem reutilizados (c) precisão nos registros de tempo dispendido nas atividades de testes (projeto, construção e execução dos Casos de Testes) por ciclo de desenvolvimento ou *sprint*.

Com o objetivo de mitigar a ameaça relacionada ao tempo necessário para experiência no uso do *template*, duas atividades são encaminhadas na etapa de preparação do estudo: (a) execução de dois *sprints* com os mesmos participantes que executaram anteriormente as atividades de projeto e construção de Casos de Testes, em *releases* anteriores - com o intuito de possibilitar um tempo de sedimentação dos conhecimentos adquiridos no uso do editor e do *template*. Sendo que os participantes devem ser os mesmos nos dois *sprints*. (b) desenvolvimento de um plano de treinamento no uso do *template* proposto, e disposição de um editor que visa facilitar observância e aderência a este *template*.

Com o objetivo de avaliar o treinamento, os participantes devem preencher um questionário sobre sua efetividade na eventualidade de um aprendizado insatisfatório, e que ações sejam encaminhadas para assegurar o entendimento do *template* e sua utilização via editor.

Quanto às eventuais inconsistências de registros de horas de trabalho registradas por atividade em cada projeto, serão coletados os lançamentos individuais registrados nas *baselines* de cada projeto, de forma que solucione eventuais inconsistências.

Para mitigar o risco de não existir quantidade significativa de Casos de Testes reutilizados nas execuções dos *sprints*, inviabilizando a possibilidade de observar oportunidades de reutilização, serão selecionados *sprints* cujas funcionalidades já foram objeto de desenvolvimento ou manutenção em *releases* anteriores do *software*.

- Validade externa:

Este estudo será executado em uma organização de *software* que utiliza um processo de testes estabelecido para um projeto de um domínio escolhido previamente. Neste sentido, seria interessante repetir o estudo em outros projetos, nas organizações que desenvolvem produtos com características distintas.

- Validade da Conclusão:

Estudos de caso são estudos observacionais que não permitem um controle mais rigoroso do contexto de execução - como é possível fazer em

experimentos controlados - sendo que, em certos casos, pode se chegar a um resultado que aponte para uma causalidade inexistente.

Correlações significativas podem ser registradas ao final da execução do estudo (especialmente quando um grande número de variáveis são medidas), ou como um resultado de uma variável "latente". Ou seja, uma variável que não foi considerada no plano do estudo e afeta uma ou mais variáveis, fazendo com que o pesquisador conclua por uma correlação inexistente (Kitchenham, 2007).

Com o intuito de mitigar inconsistências na avaliação dos resultados, um segundo pesquisador mais experiente supervisionará a execução do estudo - de forma a auxiliar no tratamento de discrepâncias em relação ao planejado, e mitigar um eventual viés de interpretação na fase de análise dos dados.

Isto porque um dos pontos relevantes na investigação é avaliar se efetivamente estas práticas contribuem para o reuso dos procedimentos de Casos de Testes. A quantidade de reuso foi definida com o objetivo de alcançar a quantidade de Casos de Testes reutilizados, atuando como *surrogate* das práticas "Visibilidade de projeto" e "Design simples".

A alternativa sugerida para alcançar uma maior agilidade nas atividades de projeto de testes foi implementar um conjunto de mudanças que visam trazer eficiência às atividades de testes - promovendo o reuso, as melhores práticas de projeto e priorizando os riscos envolvidos.

Neste contexto, talvez seja interessante repetir este estudo promovendo as mudanças sugeridas de forma gradativa, a fim de avaliar os ganhos sucessivos em que cada mudança possa trazer em termos de agilidade.

- Validade do Construto:

Para que seja possível a comparação do esforço dispendido nas atividades de projeto e construção dos Casos de Testes é necessário categorizar os Casos de Teste.

Esta classificação considera a quantidade de passos de cada procedimento de todo Caso de Teste, as ações associadas aos procedimentos de verificação das pré-condições e verificação dos resultados esperados.

Para isto, com o objetivo de mitigar eventuais inconsistências na classificação é necessária atuação de dois pesquisadores: o primeiro realiza a classificação considerando os critérios definidos e o segundo mais experiente faz uma revisão desta classificação.

Também são esperados que os profissionais que venham participar do estudo sejam os mesmos responsáveis pelas atividades de projeto e desenvolvimento dos Casos de testes, antes da introdução do *template* proposto.

5.3.3.3 Treinamento:

5.3.3.3.1 Definição e procedimento:

- O treinamento do *template* para Projeto e Construção dos Casos de Testes e da ferramenta de edição será realizado em uma sessão;
- Ao final do treinamento, os participantes preencherão um formulário de avaliação;
- Se necessário, uma nova sessão de treinamento poderá ser realizada, com o propósito de assegurar que os participantes ao iniciarem o Projeto dos Casos de Testes tenham conhecimento pleno do *template* a ser adotado.

5.3.3.3.2 Participantes:

Integrantes da equipe que participa do projeto e execução dos testes.

5.3.3.3.3 Instrutor:

Pesquisador.

5.3.3.3.4 Artefatos:

- *Template* de Casos de Testes;
- Instrução de Procedimento de Projeto de Casos de Testes;
- Slides de apresentação sobre o *template*.

5.3.3.3.5 Ferramentas:

- Editor e Visualizador de Casos de Testes.

5.3.3.4 Execução de Procedimentos:

5.3.3.4.1 Definição:

- A adoção do *template*, do editor e das mudanças sugeridas é realizada em, pelo menos, duas *sprints* executadas em um *release* do projeto escolhido.
- Evento 1:
A coleta de dados relacionados ao Evento 1 utiliza dados de execuções passadas no repositório da organização

- ✓ Levantamento do esforço de projeto e construção dispendido no desenvolvimento dos Casos de Testes dos dois últimos *releases*;
- ✓ Levantamento da quantidade de procedimentos reutilizados durante a execução dos dois últimos *releases* que não utilizaram o *template*.
- Evento 2:
Treinamento do *Template* e do Editor.
- Evento 3:
 - ✓ Esta terceira etapa ocorre durante a execução de um *Sprint* de um release do projeto escolhido;
 - ✓ Durante as atividades de projeto, construção e execução os participantes registram os esforços associados ao planejamento e execução de Testes.
- Evento 4:
 - ✓ Esta etapa final ocorre durante a execução de um segundo *Sprint* do mesmo *release* do evento 3;
 - ✓ Durante as atividades de projeto, construção e execução os participantes registram os esforços associados ao planejamento e execução de Testes.

5.4 Etapa 3: Execução

5.4.1 Preparação

As seguintes atividades foram realizadas na etapa de preparação para realizar o estudo de caso: (a) aplicação do formulário de consentimento individual para uso dos resultados do estudo por cada participante e, do formulário de caracterização do perfil dos profissionais envolvidos. (b) desenvolvimento do material de treinamento do Editor de Caso de Testes e sobre as mudanças no processo (c) desenvolvimento dos questionários de avaliação a serem aplicados ao final do estudo (d) instalação do editor de Casos de Testes em um servidor de aplicação da organização (e) seleção de procedimentos de CTs de *releases* anteriores para reescrita com adoção do *template* sugerido para avaliação de reuso.

Os formulários de consentimento e caracterização (Apêndices I e J) dos envolvidos foram preenchidos em 02/01/2013 por cinco profissionais: dois analistas de testes (um analista com experiência de apenas seis meses em projeto de testes e um analista sênior com mais de cinco anos de experiência em testes), dois testadores

iniciantes e o responsável pela equipe SEPG com boa experiência em processos de *software*.

O analista de teste de menor experiência foi o responsável pelo projeto de quase todos os casos de testes produzidos no desenvolvimento do novo *release* (execução acompanhada pelo estudo de caso), sendo poucos casos de testes planejados pelo analista sênior.

Os questionários de avaliação (Apêndice H) foram formulados, sendo consideradas as mudanças no processo e as práticas ágeis que se pretende inserir com a adoção do editor, e aplicados após encerramento do estudo de caso.

A instalação da ferramenta proposta foi realizada em 26/12/2012 em ambiente operacional da organização.

O objetivo da atividade de seleção de procedimentos de CTs de *releases* anteriores para reescrita foi selecionar rotinas que se repetiam em casos de testes distintos, sendo, neste caso, rotinas candidatas a se tornarem componentes reutilizáveis nas atividades de manutenção de CTs - quando há ocorrência de solicitações de mudanças de regra de negócio em uma funcionalidade já existente.

Para isto, antes de iniciar os eventos planejados para o estudo de Caso, o pesquisador acessou a base de dados contendo todos os artefatos de Testes de *releases* anteriores do projeto, acompanhado por um membro da equipe SEPG.

Nesta etapa, foram separados e transformados em componentes alguns procedimentos de preparação de massa de dados necessários para atendimento de pré-condições de CTs desenvolvidos originariamente, considerando a reincidência de uso de rotinas de verificação de pré-condição em vários Casos de Testes responsáveis por testar as funcionalidades desenvolvidas em *releases* anteriores. O objetivo é avaliar o nível de reuso de procedimentos de CTs de *releases* anteriores, transcritos para o novo padrão.

Contudo, os componentes criados nesta etapa não foram utilizados no estudo, pois em reunião com um dos principais clientes, o gerente do produto foi informado da necessidade de priorizar a execução de um novo módulo de funcionalidades específico para uma necessidade de negócio. Com isto, a oportunidade de observação do nível de reuso passou a abranger apenas o desenvolvimento deste módulo, e a comparação foi realizada entre o nível de reuso de procedimentos entre CTs do novo *release* que se utilizou da nova versão do processo proposto, e o reuso de procedimentos entre CTs de *releases* anteriores do *software*.

Sendo postergada para outro momento a avaliação do nível de reuso de procedimentos entre um *release* corrente e um anterior - decorrentes de manutenção de CTs para atendimento de solicitação de mudanças em funcionalidades existentes.

5.4.2 Execução do Estudo de Caso

A execução do estudo de caso compreendeu o período de 12/12/2012 a 05/05/2013. Durante estes meses foram executadas as atividades de “Especificação e planejamento do projeto” (Fase 1 – Pré-Game) e de “Análise Projeto e Construção de Software” (Fase 2 – Game) relativa aos *Sprints* 1,2 e 3.

O modelo de ciclo de vida adotado para o projeto foi incremental. Na etapa de planejamento foram executadas as atividades de elicitação e análise dos requisitos na fase 1 (pré-*game*) do processo, para definição dos incrementos/estórias a serem desenvolvidos, testados e integrados em várias iterações da fase 2, e sua posterior homologação na fase 3 – conforme previsto no processo padrão de desenvolvimento.

A extensão do período de observação, além de dois ciclos, foi necessária devido às mudanças de prioridade da área de negócios, do principal cliente do projeto, e da falta de disponibilidade de tempo do principal fornecedor de requisitos, que impõe parte das atividades de elicitação e análise ser postergadas para executá-las após a fase 1 (pré-*game*). Nos primeiros dois *sprints*, a equipe de desenvolvimento trabalhou ainda nas atividades de análise e elicitação, sendo, em alguns momentos, obrigada a compartilhar seu tempo com atividades relacionadas a outros projetos com o intuito de racionalizar os recursos da organização. O planejamento dos Casos de Testes ocorreu com mais intensidade a partir do *Sprint* 2.

Para a comparação da capacidade de reúso entre a versão do processo de testes anterior às mudanças e a versão proposta, foram recuperadas informações de dois *releases* anteriores (doravante denominadas *release A* e *release B*), com o propósito de alcançar uma amostra representativa a ser comparada com os resultados obtidos na execução do novo *release* em curso (doravante nomeada como *release C*).

Ainda nesta etapa de preparação para o estudo, retomaram-se alguns Casos de Uso (UC) e o gabarito utilizado pelos analistas - armazenado na ferramenta *Enterprise Architect* - prevendo a definição dos fluxos alternativos e de exceção, além do fluxo principal.

Como na abordagem sugerida de projeto de testes, os CTs Funcionais serão derivados de cada fluxo definido no Caso de Uso foi realizada uma avaliação preliminar de alguns UCs desenvolvidos para os *releases A* e *B*, quando foi percebido um baixo nível de detalhamento destes artefatos e, em certos casos a falta de definição de fluxos de exceção. Na figura 5-18 é mostrado um Caso de Uso da estória 14 do *Release B*, o qual utilizou a versão proposta do processo e o editor de Casos de Testes.

	UC-083 – Detalhar Perfil de Acesso
Atores	-
Pré-condições:	-
Pós-Condições:	
Objetivo	-
Fluxo Principal:	1.Usuário acessa a página de detalhes de um perfil de acesso. (RN067) 2.Sistema exibe as permissões que o perfil selecionado possui no grupo "Permissões". 3.Sistema exibe abaixo da "Descrição" as visões associadas ao perfil separado por Tipo com o título "Visões" que deve ser um link para associar uma visão ao perfil.
Fluxos Alternativos:	
Fluxos de Exceção:	

Figura 5-17 Exemplo de Caso Uso desenvolvido para Release B

Como o nível de detalhamento dos Casos de Uso pode influenciar nos esforços de projeto e construção, esta mesma avaliação foi realizada para os Casos de Uso do novo *release* acompanhado pelo estudo. E, apesar do treinamento realizado antes do início da execução do *release* C orientar sobre a importância de criação de cenários para exercitar os fluxos alternativos e de exceção, parece que esta caracterização nem sempre foi executada, dado que a relação entre Caso de Uso e Cenário de Teste em certos casos foi um para um.

5.4.2.1 Execução do Evento 1- Planejamento e Execução de CTs versão do processo sem mudanças

As atividades de planejamento e execução dos Casos de Testes que fazem parte do evento 1 abrangeram dois *releases* (A e B) do produto executados anteriormente - os quais utilizaram a versão do processo de teste original, e uso da ferramenta *TestLink*.

No âmbito do estudo, esta etapa consistiu em coletar as informações do esforço associado às atividades de planejamento e execução de testes e das ocorrências de reúso de procedimentos de Casos de Testes.

Tabela 5-17 - Evento 1 – *Releases* A e B usando versão do Processo anterior às mudanças

Identificação do Evento	Atividades	Descrição	Participantes
EVENTO 1	Projeto e Construção de CTs de Sistema para Gestão de Projetos	<i>Adoção de versão de Processo de Teste antes das mudanças propostas e uso da ferramenta TestLink.</i>	AT
	Execução de CTs de Sistema para Gestão de Projetos	<i>Adoção de versão de Processo de Teste antes das mudanças propostas e uso da ferramenta TestLink.</i>	TE
	Coleta dos esforços realizados e Procedimentos de CTs reutilizados	<i>Recuperação de dados de baselines armazenadas nos repositórios de projetos da Organização</i>	PE

5.4.2.1.1 Coleta dos dados de reúso de Procedimentos de CTs nos *releases* anteriores

As atividades de coleta e registro dos esforços realizados nas atividades de planejamento e execução de testes dos *releases*, bem como as informações a cerca de reutilização, ocorreram entre os dias 12/12/2012 e 28/12/2012, e foram acompanhadas por um membro da equipe SEPG.

A investigação de ocorrências de reúso abrange procedimentos, rotinas de verificação de pré-condições e resultados esperados. Em algumas estórias desenvolvidas, foi recuperada uma quantidade substancial de passos copiados entre diferentes procedimentos.

Uma quantidade significativa de Caso de Testes apresentam passos idênticos, que se diferencia apenas por uma ou outra ação que talvez modeladas como uma variável. Nestes casos, o reúso poderia ter ocorrido caso a equipe tivesse adotado uma abordagem de projeto em cenários, e na sequência os Casos de Testes, entradas e saídas, e procedimentos.

Esta sobreposição acontece, por exemplo, no *release* B, com os casos de Testes “CT001 - Verificar se apresenta a visão na lista Portfólio de Projetos” e “CT001 - Verificar se apresenta a visão na lista Portfólio de Proposta”, relacionados respectivamente aos Casos de Uso “Utilizar visões no portfólio de projetos” e “Utilizar visões no portfólio de propostas” cujas descrições aparecem abaixo. O conjunto de passos entre os Casos de Testes é o mesmo, com exceção do ícone escolhido (Figura 5-19).

1	Nome do CT	ID 15996: Test Case CT001 - Verificar se apresenta a visão na lista Portfólio de Proposta	2	Nome do CT	ID 15986: Test Case CT001 - CT002 - Verificar permissão pra altera a coluna na lista de portfolio
	Pré-condição	-		Pré-condição	-
	Preparação massa de dados	-		Preparação massa de dados	-
	Entradas			Entradas	
	Objetivo	-		Objetivo	-
	Passos	1- Acesse o PB com um Usuário com as seguintes permissões “Adicionar/Editar/Remover Visões publicas” e/ou Administrador do Sistema 2- Configure no perfil do usuário as visões do tipo Projeto, Componente e Questões 3- Clique em Portfólio de Proposta 4- Escolhe uma visão na combo Visão que esteja associada ao perfil do usuário		Passos	1- Acesse o PB com um Usuário com as seguintes permissões “Adicionar/Editar/Remover Visões publicas” e/ou Administrador do Sistema 2- Configure no perfil do usuário as visões do tipo Projeto, Componente e Questões 3- Clique em Portfólio de Projeto 4- Escolhe uma visão na combo Visão que esteja associada ao perfil do usuário
	Resultados esperados	Verifique que o sistema exibe a lista de acordo com as definições das checkbox e das colunas		Resultados esperados	Verifique que o sistema exibe a lista de acordo com as definições das checkbox e das colunas

Figura 5-18 Oportunidade de reúso em CTs com nomes distintos

Por vezes, acontece de dois CTs distintos terem o mesmo nome, sugerido ser responsáveis pelo mesmo conjunto de ações e, no entanto, apresentarem ações distintas.

1	Nome do CT	ID 12143: Test Case CT001 - Verificar criação do grupo "Questões" abaixo do grupo "RH"	2	Nome do CT	ID 12157: Test Case CT001 - Verificar criação do grupo "Questões" abaixo do grupo "RH"
	Pré-condição	-		Pré-condição	-
	Preparação massa de dados	-		Preparação massa de dados	-
	Entradas			Entradas	
	Objetivo	-		Objetivo	-
	Passos	1- Acessar a EAP de um projeto que tenha questões com fim real e questões sem fim real. 2- Deverá haver questões no nível do projeto, em componentes de último nível e em componentes que não sejam de último nível. 3- Clicar na aba Colunas		Passos	1-Acessar a página de foco. Deverá haver projetos que tenham questões com fim real e questões sem fim real. 2-Deverá haver questões no nível do projeto, em componentes de último nível e em componentes que não sejam de último nível. 3Deverá haver projeto/componentes com questões associadas apresentados na página de foco nos grupos "Componentes em destaque" e "Componentes selecionados" 4-Clicar na aba Colunas
	Resultados esperados	1-Verificar que em cada combo (colunas 1 a 9) foram criadas as opções "Quantidade de Questões" e "Quantidade de Questões sem fim real" no grupo "Questões", abaixo de RH, com link para lista de questões daquele componente. 2-Verificar que a quantidade de questões e de questões sem fim real está sendo apresentada corretamente.		Resultados esperados	1- Verificar que em cada combo (colunas 1 a 9) foram criadas as opções "Quantidade de Questões" e "Quantidade de Questões sem fim real" no grupo "Questões", abaixo de RH, com link para lista de questões daquele componente. 2- Verificar nos grupos "Componentes em destaque" e "Componentes selecionados".

Figura 5-19 – Oportunidade de Reúso em CTs com mesmo nome

Um exemplo é exibido na Figura 5-20 onde estão dispostos os CTS do *release* A cadastrados separadamente (Ids 12143 e 12157) e com o mesmo nome.

Em outras ocasiões, o mesmo procedimento de Caso de Teste é efetivamente reutilizado em funcionalidades distintas, porém, nestes casos, a descrição do Caso de Teste é duplicada e cada cópia relacionada separadamente a cada Caso de Teste.

1	Nome do CT	Test Case 15680: CT002 - Verificar que o usuário não tem permissão de mudar a configuração	2	Nome do CT	Test Case 15684: CT002 - Verificar que o usuário não tem permissão de mudar a configuração
	Pré-condição	-		Pré-condição	-
	Preparação massa de dados	-		Preparação massa de dados	-
	Entradas			Entradas	
	Objetivo	-		Objetivo	-
	Passos	1 - Faça login com o usuário que não tenha a permissão de Administrador de Sistema e que não tenha a permissão de "Alterar Configuração do Projeto" ou Perfil 1 no projeto 2 - Clique em Projetos		Passos	1 - Faça login com o usuário que não tenha a permissão de Administrador de Sistema e que não tenha a permissão de "Alterar Configuração do Projeto" ou Perfil 1 no projeto 2 - Clique em Projetos
	Resultados esperados	Verifique que não foi exibida a opção de Configurações em "Projetos > Configurações", o usuário não consegue acessar a página de Mudança nas Configurações dos Projetos.		Resultados esperados	Verifique que não foi exibida a opção de Configurações em "Projetos > Configurações", o usuário não consegue acessar a página de Mudança nas Configurações dos Projetos.

Figura 5-20 - Reúso de Procedimentos em CTs com mesmo nome

Este tipo de conduta dificulta a manutenção e acontece, por exemplo, no *release* B com o Caso de Teste "CT002 - Verificar que o usuário não tem permissão de mudar a configuração." associado aos Casos de Uso "UC097 - Mudar Configurações de Projetos e UC098" - Mudar Configurações de Projeto (Figura 5-21).

Para efeitos de contagem de situações como representada na figura 5-21, contabilizou-se como ocorrência de reúso. Visto que a forma como as equipes de testes dos *releases* anteriores utiliza a ferramenta de projeto dificulta a separação entre procedimento e Caso de Teste.

Nestes casos, entende-se que o analista de teste identificou o mesmo comportamento de execução e sequência de passos para Casos de Teste distintos, e não implementou esta separação à nível de procedimento pela limitação da ferramenta adotada.

5.4.2.1.2 Coleta dos esforços de testes nos *releases* anteriores

Na atividade de coleta do esforço associado às atividades de projeto e construção de CT e de execução dos testes, foram utilizados os registros de projeto dos dois *releases* A e B.

Como a equipe de teste é orientada a registrar o esforço de projeto e execução por estória, os valores homem-hora recuperados nas *baselines* destes *releases* não permitiram determinar, de maneira direta, qual foi o esforço dispendido para projetar e executar cada CT.

Através das informações obtidas no repositório da ferramenta utilizada nos *releases* anteriores contendo todos os CTs, procedimentos e quantidade total de passos de cada procedimento, e de posse dos esforços totais dispendidos para projeto e execução, foi possível calcular os esforços médios de projeto e execução de 1 passo de cada estória de cada *release*.

Para isto, foi necessário recuperar, para cada CT, o número de passos de cada procedimento, o resultado esperado e pré-condição, e obter o total geral de passos projetados para a estória (**TpCt_E**).

De posse deste total e do esforço informado de projeto de CT e esforço estimado de execução de CT foi possível obter o “Esforço Médio de Projeto de 1 Passo” e o “Esforço Médio de Execução de 1 Passo” para a estória.

$$\mathbf{EMPP_E} = \mathbf{TEP_E} / \mathbf{TpCt_E} \text{ e}$$

$$\mathbf{EMEP_E} = \mathbf{TEE_E} / \mathbf{TpCt_E}, \text{ onde:}$$

TEP_E = Esforço total dispendido nas atividades de Projeto de Teste da Estória (HH)

TEE_E = Esforço total dispendido nas atividades de Execução de Teste da Estória (HH)

TpCt_E = Soma de todos os passos dos CTs da Estória

EMPP_E = Esforço Médio de Projeto de 1 Passo da Estória (HH/Passos)

EMEP_E = Esforço Médio de Execução de 1 Passo da Estória (HH/Passos)

Com a obtenção dos valores de esforço médio $EmPp_E$ e $EMEp_E$ é possível calcular o esforço associado à atividade de projeto e o esforço associado à atividade de execução, através do produto do esforço médio da estória e do total de passos do CT.

$$EP_{CT} = EmPp_E * Tp_{CT} \text{ e } EE_{CT} = EMEp_E * Tp_{CT} \text{ onde:}$$

EP_{CT} = Esforço de Projeto do CT (HH)

EE_{CT} = Esforço de Execução do CT (HH)

Tp_{CT} = Soma de todos os passos do CT.

Este cálculo foi executado para cada estória, de cada um dos *releases* anteriores às mudanças sugeridas (A e B). Posteriormente também utilizado nas atividades de coleta de esforço para o *release* C em que foram inseridas mudanças no processo.

5.4.2.2 Execução do Evento 2 - Treinamento sobre o Editor de Casos de Testes e mudanças do processo

Foram realizadas duas sessões de treinamento do Editor. A primeira sessão ocorreu dia 02/01/2013, com a participação de um analista de teste júnior e dois testadores iniciantes. A segunda sessão sucedeu dia 06/02 para a analista de teste sênior que na ocasião do primeiro treinamento estava de férias. As sessões programadas originariamente a serem executadas em 4 horas se estenderam e o tempo de duração em ambas durou 8 horas.

Tabela 5-18 - Atividades do Evento 2

Identificação do Evento	Atividades	Descrição	Participantes
EVENTO 2	Treinamento	<i>Treinamento para uso do Template e do Editor</i>	AT, TE e PE
	Avaliação do Treinamento	<i>Aplicação de formulário de avaliação para participantes</i>	AT e TE

No treinamento, foram exibidas um conjunto de transparências com enfoque nos seguintes tópicos: (a) Mudanças realizadas no processo padrão (b) Novo *template* de projeto de Caso de Teste (c) Diretrizes para identificação e Especificação de Cenários, Casos de Testes, Procedimentos e Pré-Condições de CTs. (d) Funcionalidades da Ferramenta de Edição de Casos de Testes.

Os analistas de teste tiveram acesso à ferramenta e planejaram alguns CTs como exemplo.

Antes do início do treinamento, foram distribuídos formulários de caracterização do participante e de consentimento para a participação no estudo de caso (Apêndices I e J).

Ao final de cada uma das sessões foi distribuído formulário de avaliação. Para esta avaliação foi utilizado formulário da própria organização subdividido em cinco tópicos: (1) conteúdo, (2) dinâmica, (3) instrutor, (4) infraestrutura e (5) comentários adicionais.

O evento obteve a média geral "Ótimo", considerando os níveis Ótimo, Bom, Regular e Ruim, atendendo às expectativas dos quatro participantes. Quanto ao conteúdo, na opinião dos participantes, o treinamento alcançou os objetivos, o tema é de grande relevância, os recursos foram bem utilizados, tem contextualização do conteúdo em relação ao mercado, aplicabilidade na vida profissional e relação com a estratégia da empresa. Quanto à avaliação do instrutor, os itens obtiveram nota "Ótimo", na média e na mediana (Relacionamento, Conhecimento sobre o assunto, Administração do Tempo, Comunicação, Motivação para a Participação), a infraestrutura foi avaliada como adequada. No tópico dos comentários dos participantes, retornaram que o treinamento facilitou o entendimento de uso do editor e que seu uso contínuo pode apoiar as atividades de garantia de qualidade.

5.4.2.3 Execução do Evento 3 - Projeto e Construção de Casos de Testes com adoção das mudanças sugeridas.

O período de observação do primeiro ciclo de projeto de teste se estendeu por mais de um *Sprint* – de 04/01/2013 até 05/05/2013 - compreendendo os *Sprint* 1, 2 e 3 do *release* C. Isto porque sucessivos atrasos e adiamentos das atividades de projeto, que aguardavam definições dos *stakeholders*, fizeram com que as atividades de análise e elicitação de requisitos se estendessem até o *Sprint* 3, o que influenciou nas atividades de projeto e execução dos testes.

Tabela 5-19 Atividades Evento 3

Identificação do Evento	Atividades	Descrição	Participantes
EVENTO 3	Projeto e Construção de CTs de Sistema para Gestão de Projetos	<i>Adoção do Processo de Testes contendo mudanças sugeridas e Template baseado na IEEE-829-Primeiro Ciclo</i>	AT
	Execução de Casos de Testes de Sistema para Gestão de Projetos	<i>Adoção do Processo de testes contendo mudanças sugeridas e Template baseado na IEEE-829 – Primeiro Ciclo</i>	TE
	Coleta dos esforços realizados e Procedimentos de CTs reutilizados	-	PE

Até o momento presente - no qual estavam sendo registrados os resultados do estudo - o *Sprint #4* ainda estava em execução. Sendo assim, considerando as restrições de tempo associadas ao estudo não foi possível observar a execução de um segundo ciclo, com o intuito de gerar mais evidências a respeito dos resultados obtidos (Evento 4 do estudo de caso).

Neste terceiro evento foram observadas as atividades de planejamento e execução de nove histórias (1, 2, 3, 4, 6, 7, 8, 10, 11 e 12) do *Sprint #3*.

A atividade de “Caracterização dos Testes” se realizou na fase 1 (Pré-Game – Especialização e Planejamento do Projeto) do processo especializado SCRUM da organização, publicado na *intranet* da organização.

A caracterização dos testes se inicia nesta fase e abrange as atividades de caracterização dos itens a serem testada (quais funcionalidades farão parte do escopo de teste), definição da estratégia de testes (quais tipos de testes deveriam ser realizados).

Todavia, como inicialmente a equipe não teve acesso direto ao principal *stakeholder* que tinha conhecimento dos requisitos do projeto, sendo este substituído por um fornecedor de requisitos intermediário, muitas dúvidas surgiram no decorrer do pré-*game* fazendo com que as atividades de análise e elicitação de requisitos e caracterização de itens para teste fossem estendidas até os *Sprints #1* e *#2*.

Na priorização de quais histórias seriam desenvolvidas, o Gerente de Projeto juntamente à equipe optou por colocar as mais complexas no final do *Product Backlog*, para um momento em que o time tivesse mais conhecimento sobre os requisitos a serem desenvolvidos, visto que já era percebida uma dificuldade de acesso ao *stakeholder* principal.

Dessa forma, os casos de Uso “UC. 144 - Editar Proposta de Trabalho”, “UC. 151 - Visualizar guia Documentação”, “UC. 153 - Visualizar guia Serviços e Objetos de Custo” e “UC. 147 - Importar dados do MS Excel” avaliados como os mais complexos para desenvolvimento e testes não foram selecionados para o “*Selected Product Backlog*” dos primeiros *Sprints*. Na execução do projeto de testes foi seguida esta mesma diretriz, e o planejamento das histórias 5 e 9, associadas a estes UCs, foram postergadas para os últimos *Sprints* do projeto.

Tabela 5-20 Avaliação complexidade e importância para o Negócio dos UCs

Casos de Uso	Complexidade	Impacto (Importância e Risco para o Negócio)
UC. 136 - Visualizar menu principal	2	2
UC. 142 - Incluir Proposta de Trabalho	1	3
UC. 143 - Listar Propostas de Trabalho	2	3
UC. 144 - Editar Proposta de Trabalho	3	3
UC. 145 - Detalhar Proposta de Trabalho	2	2
UC. 146 - Excluir Proposta de Trabalho	2	2
UC. 147 - Importar dados do MS Excel	3	2
UC. 148 - Gerar Relatórios da PT	1	2
UC. 149 - Visualizar guia Gestão	1	2
UC. 150 - Visualizar guia Declaração de Escopo	1	2
UC. 151 - Visualizar guia Documentação	3	3
UC. 152 - Visualizar guia Fiscalização de Projetos	2	2
UC. 153 - Visualizar guia Serviços e Objetos de Custo	3	3
UC. 154 - Visualizar guia Histórico	1	2
UC. 155 - Visualizar topo das guias	2	1
UC. 157 - Visualizar guia Coordenação	1	2
UC. 158 - Listar arquivos excel	2	2

A estratégia de teste considerou apenas os testes unitários, de integração e funcionais. A equipe de testes cuidou apenas dos testes funcionais, ficando a cargo dos desenvolvedores os testes de unidade e integração.

Tabela 5-21 – Níveis de Teste por Caso de Uso

Caso de Uso	Funcionalidade					Usabilidade	Desempenho	Confiabilidade		
	Unid.	Integr.	Funcional	Aceitação	Sistema			Tolerância a Falhas	Segurança	Integridade de Dados
136	x	x	x	-	-	x	-	-	-	-
142	x	x	x	-	-	x	-	-	-	-
143	x	x	x	-	-	x	-	-	-	-
144	x	x	x	-	-	x	-	-	-	-
145	x	x	x	-	-	x	-	-	-	-
146	x	x	x	-	-	x	-	-	-	-
147	x	x	x	-	-	-	-	-	-	-
148	x	x	x	-	-	-	-	-	-	-
149	x	x	x	-	-	x	-	-	-	-
150	x	x	x	-	-	x	-	-	-	-
151	x	x	x	-	-	x	-	-	-	-
152	x	x	x	-	-	x	-	-	-	-
153	x	x	x	-	-	x	-	-	-	-
154	x	x	x	-	-	x	-	-	-	-
155	x	x	x	-	-	x	-	-	-	-
157	x	x	x	-	-	x	-	-	-	-
158	x	x	x	-	-	x	-	-	-	-

A identificação dos Cenários, Casos de Testes e Procedimentos foi executada a partir da atividade “Projetar Testes” concorrentemente à atividade “Realizar a especificação técnica” no *Sprint#3*.

Durante a execução desta atividade, foram identificados vinte e dois cenários, e cinquenta e quatro CTs, sendo posteriormente acrescentados de mais cento e quatorze CTs (No apêndice G está a matriz gerada nesta etapa).

Na reunião de retrospectiva do segundo *Sprint*, foi sugerido que esta identificação fosse iniciada ainda na etapa de pré-*game* com as atividades de elicitación e análise dos requisitos. O objetivo é subsidiar a estimativa de esforço de teste do projeto realizada a partir da avaliação do repositório de estimativa contendo: (a) os esforços de cada projeto desenvolvido pela organização, distribuídos por cada atividade do processo de desenvolvimento, (b) o percentual de esforço gasto em relação ao esforço total dispendido no projeto e, (c) o percentual médio da atividade, considerando todos os projetos agrupados por tipo.

De acordo com o repositório de estimativas que se utilizam do processo especializado *Scrum*, o esforço (HH) associado ao planejamento de testes funcionais representa em média 8,69% do esforço total de desenvolvimento de *software*, o esforço associado à execução dos testes funcionais representa 9,43% e o esforço relacionado à homologação 2,60%.

Para o *release C*, o esforço de planejamento de testes coletado até o encerramento do terceiro *Sprint* (66 HH) representava 4,47% do esforço total (1508 HH) realizado. E o esforço de execução de teste representava 1,93% do esforço total.

Porém, considerando um atraso nas atividades em geral, devido ao gasto de esforço não previsto nas atividades de elicitación e análise de requisitos, e que ainda estavam programados mais dois *Sprints* e a realização da fase de Homologação e Implantação (pós-*game*), estes dados não podem ser utilizados para inferir alguma conclusão a respeito do desempenho das atividades de teste deste *release*.

5.4.2.3.1 Coleta dos esforços das atividades de testes no *release C*

Para obtenção dos valores de esforço dispendido nas atividades de testes do *release C*, é empregado as mesmas fórmulas utilizadas nos *releases* anteriores, usando os dados de alocação homem-hora associados a cada uma das nove estórias, informados por cada membro da equipe para cada estória deste *release*.

Foram obtidos os valores para esforço médio de projeto e execução por passo de procedimento de CT de cada uma destas estórias. E a partir deste esforço médio, estimado esforço dispendido para planejamento e execução de cada caso de teste do *release C*. Nesta etapa os CTs foram reunidos conjuntamente em uma planilha com os CTs dos *releases* anteriores, e para cada um destes CTs a quantidade de passos existentes em cada procedimento, rotina de verificação de resultado esperado e pré-

condição, sendo assim possível estabelecer uma classificação de complexidade para cada um destes CTs.

5.4.2.3.2 Classificação da complexidade dos CTs de cada release

Para determinar a complexidade de cada CT de cada release foi seguido o critério definido na etapa de planejamento do estudo de caso, através do qual a complexidade de um Caso de Teste será calculada a partir do número de ações/passos contidos em seus procedimentos de execução do CT, preparação/verificação de pré-condição e verificação de resultados esperados.

Para recuperar as quantidades de passos de cada procedimento de cada Caso de Teste do *release* C, foram coletados os dados do repositório de dados da ferramenta de edição de CT.

Para recuperar estas mesmas quantidades dos *releases* anteriores A e B foi desenvolvida uma aplicação capaz de contar os passos de cada CT cadastrado anteriormente na ferramenta *TestLink*.

Ao final deste trabalho, foi possível reunir as quantidades de passos dos 499 CTs relativos aos três *releases* (A, B e C) de forma a apoiar a classificação de complexidade destes artefatos.

Para definição de qual melhor medida a ser utilizada na definição das faixas de complexidade foi verificada a normalidade da distribuição das quantidades de linhas de cada CT.

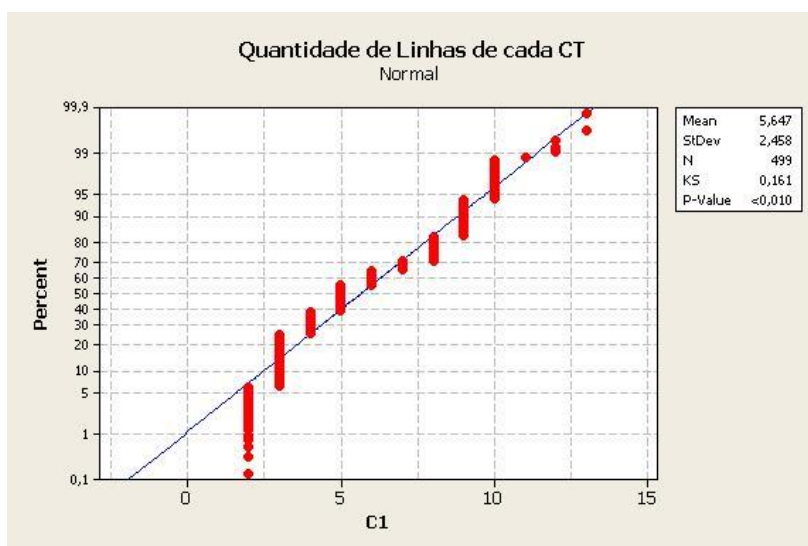


Figura 5-21 – Avaliação normalidade da distribuição de Qtde de Linhas dos CTs produzidos

O teste selecionado para verificar a normalidade da distribuição contendo a quantidade total de linhas de cada Caso de Teste dos três *releases* foi o Kolmogorov-Smirnov. Na execução deste teste, o *p-value* obtido foi melhor do que 0.01 e, por conseguinte, verificou-se que a distribuição não era normal (Figura 5-22).

Utilizando a mediana (5), o primeiro quartil (3) e o terceiro quartil (8) foram definidos três faixas de complexidade com os seguintes intervalos: alta (Casos de Testes cuja quantidade de linhas é maior ou igual ao terceiro quartil), média (Casos de Testes cuja quantidade de linhas é menor que o terceiro quartil e maior que o primeiro quartil) e baixa (Casos de Testes cuja quantidade de linhas é menor ou igual ao primeiro quartil).

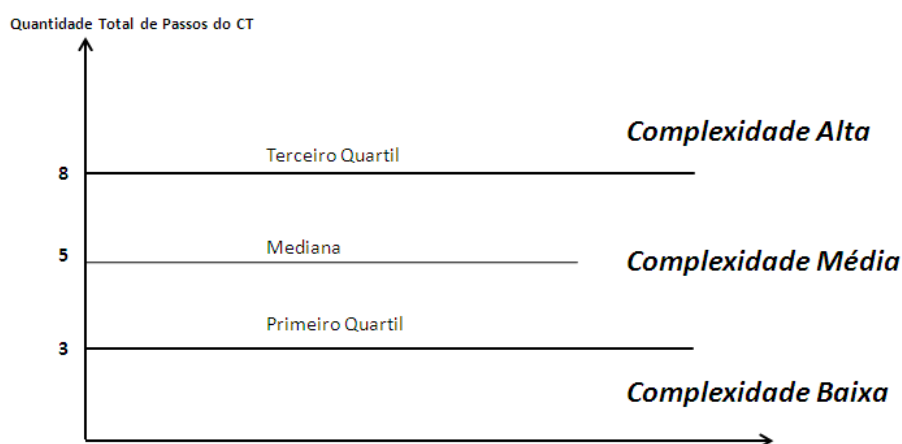


Figura 5-22 – Avaliação do nível de complexidade dos Casos de Teste de cada *Release*

Com esta escala ordinal foram classificados os Casos de Teste de cada *release* e obtido um somatório da incidência destas faixas para cada um dos *releases* (Figura 5-21).

Considerando o critério definido para as faixas de complexidade de CT, nenhum dos Casos de Teste do *release* C foi categorizado como Baixa Complexidade.

Tabela 5-22 - Classificação Casos de Testes *Releases* A e B

Totalizadores	Release A	Release B	Baseline (Consolidado das Releases A e B)	Release C
Qtde Total de CTs produzidos	231	100	331	168
Qtde CTs complexidade baixa:	103	22	125	0
Qtde CTs complexidade média	117	68	185	41
Qtde CTs complexidade Alta	11	10	21	127

A tabela 5-23 mostra os totais de CTs produzidos em cada *release* e os esforços totais dispendidos nas atividades de planejamento e execução de CTs.

Tabela 5-23 - Quantidade de CTs dos *releases* e Esforços associados.

Totalizadores	Release A	Release B	Baseline (Consolidado das Releases A e B)	Release C
Esforço HH dispendidas na Atividade de Projeto e Construção:	60,00	33,34	93,34	65,90
Esforço HH dispendidas na Atividade de Execução:	29,42	19,29	48,71	29,33
Somatório de linhas dos Casos de Testes:	987	488	1475	1341

5.4.2.3.3 Coleta dos dados de reúso Procedimentos de CTs do novo *release*

O reaproveitamento de trechos de rotinas e componentes no *release* C ocorreu tanto em relação aos procedimentos de execução do Caso de Testes quanto nos procedimentos associados às pré-condições. Neste caso, os analistas de teste descreveram nas rotinas de pré-condição, os passos necessários para atender o estado obrigatório das condições do teste.

Para recuperar estas informações, foi acessada a base relacional da ferramenta (EditorCT) - repositório dos procedimentos e casos de teste planejados, e dos registros de execução.

Da mesma forma que na coleta dos dados de reúso de procedimentos para os *releases* A e B, cada ocorrência de reúso recuperada foi inserida em uma planilha para contagem e comparação.

5.5 Etapa 4: Empacotamento

5.5.1 Avaliação e Análise

5.5.1.1 Questão 1 – Reutilização de Procedimentos

Para quantificar o nível de reúso de procedimentos, foi considerado, neste cálculo, cada reúso de procedimento, pré-condição ou resultado esperado pelos Casos de Testes classificados por complexidade.

Entretanto, como não ocorreu reúso de rotina de resultado esperado para os três *releases* que participaram do estudo, o total de ocorrências de reutilização (ToRe) por complexidade foi obtido a partir apenas da soma das ocorrências de reúso de procedimento de execução de CT e reúso de rotinas de pré-condição.

$$\text{ToRe}_{\text{Comp}} = \text{ToRePe}_{\text{Comp}} + \text{ToRePp}_{\text{Comp}}, \text{ onde:}$$

ToRe_{Comp} = Total de ocorrências de reúso em CTs por complexidade

ToRePe_{Comp} = Total de ocorrências de reúso de procedimentos de execução em CTs por complexidade

ToRePp_{Comp} = Total de ocorrências de reúso de procedimentos de pré-condição em CTs por complexidade.

comp = Podendo assumir os valores baixa, média e alta.

Com a obtenção destes totais e as informações de totais de Casos de Testes por complexidade e por *release*, foi calculada a taxa de incidência de reúso por complexidade de CT (**TxRe_{Comp}**), resultado da divisão do total de ocorrências de reutilização (**ToRe_{Comp}**) por complexidade de Caso de Teste pelo número total de Casos de Testes classificados por complexidade (**ToCT_{Comp}**):

$$\mathbf{TxRe_{Comp} = ToRe_{Comp} / ToCT_{Comp}}$$

Este procedimento foi realizado para cada categoria de complexidade de Caso de Teste de cada uma dos *releases*. Sendo que no *release* C, como para uma quantidade substancial de CTs houve reúso tanto do procedimento de execução, como reúso de rotina de pré-condição, os valores obtidos para as taxas de reutilização de média e alta complexidade foram maiores do que um (Tabela 5-24).

Tabela 5-24 - Taxa de Reúso de Procedimentos em cada *Release*

Release	Categoria Complexidade	Total de Caso de Testes	Reutilizações de CTS usando mesmo procedimento:	Reutilizações de CTS usando mesma rotina pré-condição:	Total Reutilizações	Reutilizações/ Total de Casos de Testes
A	Baixa Complexidade	103	36	0	36	0
	Média Complexidade	117	2	0	2	0,02
	Alta Complexidade	11	0	0	0	0,00
B	Baixa Complexidade	22	5	0	5	0,23
	Média Complexidade	68	9	0	9	0,13
	Alta Complexidade	10	0	0	0	0,00
C	Baixa Complexidade	0	0	0	0	0
	Média Complexidade	41	37	39	76	1,85
	Alta Complexidade	127	119	127	246	1,94

Em relação à capacidade de reúso em CTs de baixa complexidade, como não houve CTs classificados nesta categoria para o *release* C, não é possível refutar a hipótese nula, através da qual não existiria diferença significativa entre a taxa de ocorrência de reúso entre projetos de testes que se utilizam das mudanças propostas e aqueles que se utilizaram da versão de processo de teste anterior a estas mudanças (Hipótese **H_{01_1}**).

Contudo, ao comparar as taxas de reúso em CTs de média e alta complexidade obtidas na execução do *release* C (respectivamente 1,85 e 1,94), os valores envolvidos sinalizam que a adoção destas mudanças pode efetivamente promover a reutilização de artefatos de testes, sugerindo que o aumento de visibilidade de projeto e o fornecimento de suporte para promover a prática de *design* simples podem favorecer o reúso dos procedimentos (Hipóteses **H_{11_2}** e **H_{11_3}**).

5.5.1.2 Questão 2 – Esforço dispendido em Projeto e Construção de CT

Com a classificação de complexidade de cada Caso de Teste envolvido no estudo e com a obtenção dos esforços dispendidos de projeto para cada caso de teste classificado, foi possível obter o esforço médio de projeto e esforço médio de execução por complexidade de CT de cada *Release*.

Tabela 5-25 - Esforço Médio de Projeto de CT por release

Médias	Release A	Release B	Baseline (Consolidado das Releases A e B)	Release C
Esforço médio para Projetar CTs de Baixa Complexidade (Min.)	10,22	13,80	10,85	-
Esforço médio para Projetar CTs de Média Complexidade (Min.)	18,56	20,41	19,24	16,11
Esforço médio para Projetar CTs de Alta Complexidade (Min.)	34,52	33,61	34,08	25,98

Foi calculada a média aritmética dos esforços de projeto do conjunto de CTs agrupados por complexidade de cada *release*. Estes valores aparecem na tabela 5-25.

Como nenhum dos Casos de Testes que adotaram as mudanças de projeto foi classificado como de baixa complexidade, não foi possível determinar se o esforço dispendido nas atividades de Projeto e Construção dos CTs de Baixa Complexidade apresenta diferença significativa em equipes que adotaram o processo modificado e o *template* proposto. Sendo assim, a hipótese nula associada à questão R2.1 não foi refutada (Hipótese **H₀2_1**).

Quanto à mesma comparação em relação ao esforço dispendido de projeto de CTs de média complexidade (questão R2.2), o tempo médio de projeto de casos do *release C* (16,11 min) apresentou um valor mais baixo do que a mesma métrica obtida para os Casos de Teste do *release B* (20,41 min) e os Casos de Testes do *release A* (18,56). Sendo assim, para este conjunto de dados a hipótese 2 da questão R2.2, foi avaliaram como verdade (Hipótese **H₂2_2**).

Da mesma forma, ao examinarmos os esforços médios dispendidos para Casos de Testes de alta complexidade, o esforço médio calculado para o projeto de CTs do *release C* apresenta um valor significativamente mais baixo (25,98) se comparado com os esforços médios do *release a* (34,52) e B (33,61), representando um ganho substancial (Hipótese **H₂2_3**). Podendo ser parte de estes ganhos decorrentes do fato do mesmo componente de pré-condição ter sido associado a mais de oitenta e cinco CTs. Isto porque, ao incluir os passos deste componente como parte do esforço de projeto, o total de passos de cada um dos CTs aumentou, a ponto de

serem classificados como alta complexidade, apesar de possuírem uma quantidade menor de passos de procedimentos de execução.

Outras questões podem ter influenciado este resultado: (a) como não foi possível acompanhar todos os *sprints* do *release* que utilizou as mudanças sugeridas, as histórias de maior complexidade, postergadas para seleção nos últimos *sprints* ainda não tinham sido executadas quando o estudo de caso foi encerrado. (b) os esforços de retrabalho nos CTs decorrente de mudanças de requisitos não são contabilizados separadamente e também podem constituir um viés a ser avaliado.

Porém, de todo modo, se considerarmos a perspectiva exclusiva de produção de passos de procedimentos, esta estratégia de reúso pode mitigar esforços em longo prazo, com a geração de uma biblioteca de componentes para reúso pelo analista de testes.

5.5.1.3 Questão 3 – Esforço dispendido na Atividade de Execução de CT

Da mesma forma que na comparação do esforço médio dispendido nas atividades de projeto e construção de CT, foi possível comparar apenas os esforços médios de execução de testes dos CTs de média e alta complexidade. E neste caso, a hipótese nula **H_{03_1}** associada ao esforço dispendido nas atividades de execução de CTs de baixa complexidade não foi refutada.

Tabela 5-26 - Esforço Médio de Projeto de CT por *release*

Médias	Release A	Release B	Baseline (Consolidado das Releases A e B)	Release C
Esforço médio para Executar CTs de Baixa Complexidade (Min.)	4,16	6,62	4,59	-
Esforço médio para Executar CTs de Média Complexidade (Min.)	9,67	13,03	10,90	7,50
Esforço médio para Executar CTs de Alta Complexidade (Min.)	17,76	13,52	15,74	12,41
Quantidade de CTs do Release	231	100	331	168
Quantidade de defeitos	14	-	-	35

Em relação à CTs de média complexidade, existe um ganho no tempo de execução ao compararmos o esforço médio de execução de CTs de média complexidade do *release* C (7,50 min) em relação ao esforço médio de execução de casos do mesmo tipo para o *release* B (13,03 min), como também em relação ao esforço médio obtido no *release* A (9,67 min). Sendo assim, para os dados observados a hipótese 2 associada à questão R3.1, avaliaram como verdade. (Hipótese **H_{23_2}**)

Em relação aos CTs de alta complexidade, parece existir um ganho significativo se considerarmos o esforço médio dispendido pelas atividades de execução do *release* C (12,41) se comparados com os esforços médios dos *releases* A e B (17,76 e 13,52) (Hipótese **H_{13_3}**).

Uma ressalva a ser registrada diz respeito a uma melhor relação entre quantidade de CTs executados por quantidade de defeitos revelados no *release* C, se comparada com a mesma taxa no *release* A (devido a um problema do ambiente de armazenamento dos dados das *baselines* não foi possível recuperar as informações de defeitos relativas ao *release* B).

Esta diferença não pode ser diretamente associada a uma maior eficiência de detecção de erros por conta de um melhor projeto de teste, já que a equipe de desenvolvimento escalada para a construção do *software* apresentava menos maturidade e experiência do que a equipe de desenvolvimento do *release* A.

5.5.1.4 Avaliação dos participantes

Após a finalização da coleta dos dados, foram aplicados dois formulários de avaliação para participantes da: (a) equipe de testes (b) equipe SEPG responsável pela evolução do processo de *software* da organização. (Apêndice H)

5.5.1.4.1 Mudanças no Processo de Testes

Dois participantes do SEPG preencheram o formulário registrando percepções acerca da inclusão de novas atividades e mudanças sugeridas para o processo de testes.

Em relação à inserção da atividade “Caracterizar testes e Planejar Estratégia” na primeira fase do processo de desenvolvimento, os dois respondentes informaram que de certa forma esta atividade já era realizada, porém não era formalizada.

Segundo um dos responsáveis pelo SEPG a caracterização promoveu a antecipação das atividades de planejamento e pode apoiar a estimativa dos esforços associados às atividades de planejamento e execução de testes. Atualmente, esta estimativa é elaborada a partir das informações de um repositório contendo os esforços realizados e estimados de cada projeto da organização categorizados por cada uma das atividades dos processos definidos de desenvolvimento de *software*.

Quanto à antecipação da atividade “Planejar Testes” para ser executada concorrentemente às atividades de “Preparação da Integração do Produto” e “Especificação Técnica” na fase de “Análise, Projeto e Construção do *Software*” o gerente dos processos de Verificação e Validação acredita que pode trazer mais

agilidade ao projeto, tendo em vista que, logo após a geração e validação do Documento de Requisitos, o planejamento dos testes pode ser iniciado. Já o responsável pelo Grupo de Processos entende que apesar de trazer agilidade, devido à equipe de testes reduzida e ter atuação paralela nos projetos, esta antecipação obrigará o cronograma de cada projeto ser bem refinado e respeitado, pois com o maior envolvimento da equipe de teste, a disponibilidade para a alocação paralela deve diminuir ao menos no momento inicial.

Em relação à inserção de atividade “Projetar testes do Produto” antes da atividade atual “Planejar os testes do Produto” na fase de “Análise, Projeto e Construção do *Software*” ambos concordaram que é uma atividade que é capaz de promover um melhor projeto de Testes, pelo fato de ajudar na priorização e identificação prévia dos CTs. Um dos respondentes registrou a intenção do SEPG de modelar as atividades de identificação dos CTs após a Caracterização e Planejamento da Estratégia, ainda na Fase 1, pois poderá contribuir para o refinamento das estimativas das atividades de teste.

Quanto à adoção do Editor de Casos de Testes e do *Template*, um dos participantes respondeu que “A adoção do Editor de Casos de Teste contribuiu fortemente para o entendimento dos conceitos e definições das boas práticas para planejamento e realização de testes, visto que propicia uma melhor organização das atividades de teste.” Ainda em relação ao Editor, sugeriu como melhoria o fornecimento de mais recursos para facilitar o acompanhamento da execução dos testes, como gráficos e indicadores de andamento.

O segundo respondente afirmou que o uso do *template* e do Editor proposto permitirá a reutilização dos itens dos CTs em outras versões do produto, aumentando a produtividade no planejamento dos testes, além de orientar o preenchimento correto (resultados esperados, pré-condições, variáveis e procedimentos de execução).

Contudo, ao serem indagados, se observaram um ganho de agilidade com a adoção das mudanças, ambos responderam que pelo fato da equipe não possuir alta maturidade, os ganhos irão ocorrer mais à frente, quando a equipe estiver mais capacitada no uso do editor e das atividades sugeridas. Um deles, apesar de creditar alguns dos sucessos de reúso e organização dos testes ao editor não percebe um ganho imediato exclusivamente se a organização utilizasse apenas o *template*.

5.5.1.4.2 Utilização do Editor

O formulário de avaliação do Editor de Casos de Testes foi preenchido por dois profissionais que atuaram como analista de teste e testador na equipe do projeto do

release C. Sendo que um mesmo profissional atuou tanto como analista quanto testador, e outro apenas como analista. Neste formulário, questionaram o impacto decorrente do uso do Editor para criação e manutenção de cada elemento envolvido no Projeto de Casos de Testes.

Em relação à criação de Casos de Testes e procedimentos, um dos analistas respondeu que o uso do Editor facilitou a identificação dos CTs e a separação de cada passo e da disposição de comandos simplificou a interface para o usuário editar os procedimentos. O outro respondente contrapôs que a codificação de cada passo separadamente demanda um tempo de adequação pela equipe e costume.

Em relação à separação dos passos da pré-condição, um dos analistas que mais utilizou este recurso destacou este ponto como positivo, e outro respondente contesta que não percebeu diferença significativa.

Quanto ao uso das ACLs, ambos assinalaram como um recurso útil. Um dos respondentes registrou a necessidade de implementar uma rotina para detalhamento do conteúdo do componente na função de impressão do plano de testes. Atualmente, o plano de testes gerados faz referência ao componente utilizado, porém sem detalhar seu conteúdo.

Quanto ao uso de mapas para visualização da rastreabilidade, um dos respondentes considerou bastante útil para manutenção dos procedimentos dos CTs a possibilidade de recuperar quais Casos de Testes reutiliza um determinado procedimento. O que parece ser um ganho em relação à ferramenta utilizada anteriormente, que possuía rastreabilidade apenas em um sentido: a partir dos requisitos, passando por Casos de Usos, Casos de Testes até chegar a procedimentos.

Uma lista de novas funcionalidades foi sugerida sendo anexadas a uma *to-do list* criada na etapa de desenvolvimento do EditorCT (Apêndice L).

Em relação às mudanças no processo, um dos analistas de testes sugeriu a execução de novos *Sprints* utilizando a nova versão de processo, para que fosse possível obter um diagnóstico mais preciso. E que o uso do Editor facilitou o projeto de testes em geral. Ambos apontaram o reúso de componentes como um recurso que fornece agilidade às atividades de projeto.

5.5.2 Ameaças a Validade

Alguns fatores de riscos elencados na etapa de mapeamento das ameaças de validade interna no planejamento do Estudo de Caso não puderam ser mitigados: (a) em consequência de adiamentos solicitados pelos *stakeholders* não foi possível estender o período de observação por dois ciclos de execução dos Casos de Testes

para permitir que a equipe do *release* ganhasse fluência no *template* e na ferramenta proposta (b) este atraso também fez com que os profissionais originariamente selecionados, para participar do estudo, foram alocados em outros projetos. Quando a execução do projeto foi autorizada uma equipe de desenvolvimento com baixa maturidade foi escalada para o desenvolvimento. (c) a alocação dos esforços dispendidos pela organização, além de realizados por estória, não distinguem esforço associado ao projeto de testes daquele decorrente de mudanças de requisitos e nem tampouco o esforço gasto em teste do esforço de re-teste - resultado de uma correção de defeito encontrado.

Outro ponto a ser considerado é a quantidade de participantes envolvidos no estudo: (a) pela equipe de desenvolvimento um analista de sistema, dois desenvolvedores, dois analistas de teste (um deles *part-time* e outro acumulando as atividades de testador), e (b) pela equipe do SEPG, os responsáveis pela manutenção do processo de *software* da organização (oito profissionais, ao todo). Como a organização é de pequeno porte não foi possível definir outros arranjos em que outros profissionais pudessem ser envolvidos. Esta carência de recursos também refletiu em certos momentos no compartilhamento de um recurso por mais de um projeto.

Apesar de estes fatores trazerem um viés de interpretação em relação aos resultados obtidos, as respostas dos participantes aos questionários sinalizando sobre o ganho de organização e clareza na definição dos procedimentos, além da ocorrência de reúso de alguns procedimentos, reforçam a possibilidade de se obter ganhos decorrentes da promoção de Visibilidade de Projeto e *Design Simple* nas atividades de testes de *software*.

5.6 Conclusão.

Neste capítulo, foi apresentada a execução dos procedimentos modelados para apoiar a inserção de agilidade em processos de Testes.

Durante a execução destes procedimentos, foi planejado e executado um estudo de caso com o objetivo de investigar possíveis ganhos de reúso e esforço, decorrentes da inserção de práticas ágeis no contexto das atividades de testes, em uma organização que possui um processo definido para desenvolvimento de *software*.

Os resultados obtidos fortalecem a importância de prover mecanismos de reúso de artefatos de testes para racionalizar os esforços dispendidos em atividades de planejamento e execução dos testes. A promoção de maior visibilidade destes artefatos, para compartilhamento entre os membros da equipe, e o incentivo da prática

de *design* simples favorece a obtenção de um projeto de teste capaz de trazer mais acurácia e adequação às funcionalidades do fornecido ao cliente.

Como a organização pretende dar continuidade ao processo de melhoria das atividades de testes, novos estudos de observação serão executados para observar os novos ajustes sugeridos para o processo, como também para avaliar as novas funcionalidades as serem implementadas na ferramenta de edição.

6 Conclusão e Trabalhos Futuros

Neste capítulo, estão apresentadas as conclusões desta dissertação, resumindo a proposta da pesquisa, sua motivação e contribuições. Com o intuito de indicar possibilidades de desdobramentos e direções a serem tomadas, são sugeridos os trabalhos futuros.

6.1 Considerações Finais

Nesta dissertação, foram investigadas formas de propor adaptações em atividades de um processo de testes de uma organização, com o objetivo de acrescentar características de agilidade que porventura não estivessem sendo observadas na execução destas atividades, e cuja adoção pudesse efetivamente contribuir para melhoria do processo.

Para isto, foram empreendidas buscas na literatura técnica e projetado um estudo secundário com vistas a coletar evidência de especialização de processos de testes visando alcançar estes objetivos. A razão deste primeiro estudo foi agregar conhecimento para as mudanças a serem realizadas no processo de testes de organizações candidatas a utilizarem o conjunto de procedimentos modelado durante a pesquisa.

A abordagem foi baseada em um processo formal, desde que atividades de testes devem ser documentadas, e o acesso aos artefatos produzidos deve ser facilitado para que os membros da equipe possam visualizar seu conteúdo de forma rápida e direta. Isto porque é necessário assegurar que um membro da equipe de testes possa reexecutar estas atividades (projeto, desenvolvimento e execução) a qualquer momento que se faça necessário, mesmo não sendo o responsável pela criação do artefato.

Na etapa de avaliação do processo de testes da organização selecionada para a pesquisa, foram consideradas as restrições e aspectos do contexto (método de trabalho, nível de maturidade e dificuldades reportadas). Por conseguinte, algumas das práticas descritas na abordagem proposta não foram inseridas no escopo do estudo de caso realizado, sendo priorizadas aquelas cuja presença poderia contribuir de forma efetiva com melhoria nas partes mais críticas do processo de testes da organização.

Para investigar sobre eventuais *gaps* e propor algumas mudanças no processo de teste, um conjunto de procedimentos foi definido e executado conjuntamente com o grupo de processos da organização. A partir do resultado desta execução, foi selecionado um subconjunto de adaptações e ajustes a serem implementados.

No desdobramento deste diagnóstico, foi vislumbrada a oportunidade de projetar e construir uma ferramenta com o objetivo de apoiar a atividade de projeto de Casos de Teste.

Posteriormente, foram elicitados requisitos funcionais para execução e registro de atividades de testes, de forma a permitir que através do uso de um único *software*, a equipe de testes pudesse realizar as atividades de análise, construção, execução e registro de execução dos Casos de Testes projetados.

Um estudo de caso foi planejado e executado com o objetivo de avaliar a inserção da prática de *design* simples e visibilidade de projeto nas atividades de Projetar Testes, Especificar Casos de Testes e Definir Procedimentos de Teste.

Os resultados deste estudo estão publicados no capítulo cinco desta dissertação e apontam para a possibilidade de ganhos reais de reúso e diminuição de esforços, a partir da implementação das mudanças no processo de testes e no uso de um *template* com apoio de uma ferramenta que facilita sua adoção.

Os participantes relataram que o uso de componentes permitiu encapsular as sequências de passos de navegação entre as telas do *software* sob teste, sendo que, anteriormente, elas eram copiadas e repetidas em cada CT, e os mapas de rastreabilidade e uso de cenários facilitaram a organização e recuperação dos CTs.

Foi incorporada pela organização a forma de documentação dos processos de desenvolvimento e testes da organização (através da publicação em ambiente de Intranet e mantida a partir do uso da especificação SPEM).

Existem planos de uso da ferramenta de Edição de Casos de Testes no desenvolvimento de outros *releases* do produto selecionado para execução do estudo de caso.

As mudanças no processo até a finalização desta dissertação estavam sendo avaliadas com vias as serem incorporadas nas atividades de desenvolvimento e testes.

Concluindo esta etapa final é interessante revelar a preocupação em todo o decorrer deste trabalho, em traduzir quais características poderiam ser consideradas ágeis no contexto da execução das atividades de teste.

Como posto por Conboy, o conceito de agilidade não é facilmente transmitido e compreendido. O que parecia ser facilmente entendido, logo após a publicação do manifesto ágil, se mostrou pouco inteligível quando artigos começaram a se utilizar

dos termos agilidade e método ágil em contextos distintos com o objetivo de expressar o mesmo significado.

Agilidade é um conceito altamente multifacetado e tem sido usado por diferentes pessoas para se referir a fenômenos distintos.

Afirmar que um determinado método é ou não é ágil é quase sem sentido, dada à falta de consenso sobre o que o termo "ágil" se refere (CONBOY, 2009).

Foi a partir desta perspectiva, da leitura da pesquisa sobre características e práticas ágeis conduzida por Abrantes (2012), que nasceu a motivação para a condução deste trabalho.

Neste cenário, o estabelecimento de novos estudos secundários sobre o tema podem gerar evidências que permitam maior entendimento dos fenômenos associados à execução das atividades de teste.

6.2 Contribuições da Pesquisa

As contribuições decorrentes do desenvolvimento desta pesquisa são:

- Desenvolvimento de protocolo de uma *quasi-revisão* sistemática que visa recuperar relatos de estudos primários descrevendo a adoção de características de agilidade em processos de testes de *software*;
- Implementação de um conjunto de procedimentos visando apoiar a melhoria das atividades de testes de uma organização e inserção de práticas ágeis;
- Desenvolvimento de um protocolo de estudo primário de caráter exploratório visando observar a adoção das práticas ágeis *Design Simples* e *Visibilidade de Projeto* em um conjunto de atividades de testes;
- O relato de experiência de um estudo de Engenharia de *Software* Baseado em Evidência em uma organização selecionada para este fim;
- Desenvolvimento de um editor de Casos de Testes que visa apoiar o projetista na observância das melhores práticas de projeto de teste.

6.3 Limitações

As principais limitações identificadas nessa pesquisa se referem a:

- Execução do conjunto de procedimentos elaborado para mapeamento das atividades de testes padrão e as atividades do processo de testes da organização selecionada, foram facilitadas pelo fato desta organização apresentar um nível de maturidade no acompanhamento das atividades de melhoria de processo e também pelo fato da empresa ser de pequeno porte

permitindo que a comunicação ocorresse de forma bastante satisfatória. Neste sentido seria interessante repetir este estudo para outras empresas de maior porte, que possuíssem um processo de testes formalizado e com características distintas;

- O escopo inicial do *release* do projeto selecionado para o Estudo de Caso consistiu em funcionalidades relacionadas às funções já existentes no produto. No decorrer da negociação com o cliente, foi priorizado um *release* para implantação de um novo módulo, com características bastante distintas do escopo das funcionalidades principais do *software*.

Com isto, algumas possibilidades de reuso e visualização de blocos de procedimentos de casos de testes ficaram um pouco comprometidas. Para melhor avaliar os efeitos destas intervenções, a organização se mostrou interessada em utilizar a ferramenta nos próximos *sprints*, dedicados ao desenvolvimento de novas versões do produto.

6.4 Questões em Aberto

No contexto da pesquisa apresentada nesta dissertação, algumas questões permanecem em aberto, e conseqüentemente candidatas a futuras investigações:

- Em projetos com características distintas apresentadas na pesquisa, conduzidos por outras organizações com baixo nível de formalismo em suas atividades de desenvolvimento, a abordagem sugerida de planejamento, *design* e monitoramento das atividades poderá apresentar os mesmos ganhos obtidos nesta pesquisa?
- As facilidades de visualização e recuperação dos componentes desenvolvidos através da ferramenta de Casos de Testes são capazes de favorecer o reuso destes artefatos, mesmo em grandes projetos com uma gama de requisitos bem maiores do que os requisitos existentes no projeto estudado na organização?

6.5 Trabalhos Futuros

- A pesquisa abordou apenas a adoção da prática *Design* Simples e Visibilidade de Projeto nas atividades de teste, sendo interessante dar continuidade a estudos que utilizam outras práticas;
- A re-execução do protocolo de *quasi-revisão* sistemática sobre características de agilidade em processos de testes pode trazer novos

resultados e descobertas sobre os desafios de trazer agilidade às atividades de teste;

- Ao realizar o mapeamento das atividades de testes, utilizaram uma perspectiva de transformação, considerando as atividades, entradas, saídas e artefatos. Entretanto, na etapa de revisão da literatura técnica, coletaram relatos de adoção de um modelo orientado a fluxo que as passagens do produto pelas etapas do ciclo de vida (*hand-offs*) e os gargalos são visualizados e monitorados para melhorar o *cycle-time* do processo. É interessante considerar o projeto e execução de um estudo, a fim de acompanhar o fluxo de atividades entre um processo de desenvolvimento, conduzido por uma equipe à parte e um processo de testes em grandes organizações. Dependendo dos resultados obtidos, novas evidências seriam agregadas ao estado atual da pesquisa da prática *Kanban*;
- A repetição dos procedimentos de mapeamento das atividades de testes e a execução do estudo de caso com adoção da ferramenta de edição de Casos de Testes em outras organizações possibilita gerar novas evidências que fortaleçam a abordagem, indicando novas direções para a pesquisa de adoção de características de agilidade em processos de teste.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABRAN, A., BOURQUE, P., DUPUIS, R., MOORE, J. W., "Guide to the Software Engineering Body of Knowledge – SWEBOK", IEEE Press, Piscataway, NJ, 2001.
- ABRANTES J. F., (2012) "Práticas e Características de Agilidade em Processos de Testes de Software", In UFRJ Tese de Doutorado, 2012.
- ANDERSON, D. J., (2010) "Successful Evolutionary Change for Your Technology Business" Blue Hole Press 2010
- ANDREA, J., (2007) "Envisioning the Next Generation of Functional Testing Tools" in IEEE Software, 2007
- ALEXANDER, C., (1979) "The timeless way of building", Oxford University Press, New York, 1979
- AOYAMA, M. (1998), "Agile software process and its experience", 20th international conference on software engineering, 1998
- ARAÚJO, R.; CAPELLI, C.; GOMES, JR, A. G.; PEREIRA, M.; IENDRIKE, H.S.; IELPO, D.; TOVAR, J. A. (2004) "A Definição de Processos de Software sob o ponto de vista da Gestão de Processos de Negócio". Artigo publicado no VI Simpósio Internacional de Melhoria de Processos de Software (SIMPROS) Departamento de Informática Aplicada/UNIRIO, Rio de Janeiro, Brasil.
- BAHLI, B., ZEID E., (2005) "The role of knowledge creation in adopting extreme programming model: an empirical study", in: ITI 3rd International Conference on Information and Communications Technology: Enabling Technologies for the New Knowledge Society, 2005.
- BASILI, V., CALDIERA, G., MCGARRY, F., PAJERSKY, R., PAGE, G., WALIGORA, S., (1992) The Software Engineering Laboratory an operational software experience factory, ACM, 1992
- BENDRAOU R., COMBEMALE B., CREGUT X., GERVAIS M. (2007) "Definition of an executable SPEM 2.0" , Laboratoire d'Informatique de Paris 6, Paris, France et Institut de Recherche en Informatique de Toulouse, Toulouse, France
- BECK K. (1999) "Extreme Programming Explained: Embrace Change." Addison-Wesley

- BELL, T.E., THAYER, T.A. (1976). "Software requirements: are they really a problem?"
2nd International Software Engineering Conference, October 1976.
- BOURQUE, B. P. "Fundamental principles of software engineering—A journey", *Journal of Systems and Software*, 62 (1), p. 59, 2002.
- BIOLCHINI, M., J., P.G., NATALI, A.C., TRAVASSOS, G.H., "Systematic Review in Software Engineering: Relevance and Utility", In Technical Report. PESC-COPPE/UFRJ. Brazil. <http://www.cos.ufrj.br/uploadfiles/es67905.pdf>.
- BOEHM, B. (1976) "Seven basic principles of software engineering", State of The Art Report on SE Techniques, Infotech International Ltd., Mmdenhead, UK, 1976.
- BOEHM, B. (1985) "A Spiral Model of Software Development and Enhancement" Proc. International Workshop Software Process and Software Environments, ACM Press, 1985.
- BOEHM, B., TURNER, R. (2009). "Balancing Agility and Discipline - A Guide for the Perplexed", Addison-Wesley, 2009
- BLACK, R. (2002). "Managing the Testing Process", p415. Wiley 2002.
- BROOKS, F. (1995). "The Mythical Man Month: Essays on Software Engineering. Reading", MA: Addison-Wesley
- BURSTEIN, I., (2003). "Practical Software Testing", Springer Science Cap 8.
- BRIAND L., LABICHE Y. (2004) "Empirical Studies of Software Testing Techniques: Challenges, Practical Strategies, and Future Research" In ACM SIGSOFT Software Engineering Notes, 2004
- CANGUSSU , J. W., "A control approach for agile processes", In Computer Software and Applications Conference, 29th Annual International, Scotland, UK, 2005.
- CAUSEVIC, A., SUNDMARK, D., PUNNEKKAT S., "An Industrial Survey on Contemporary Aspects of Software Testing", 2010 Third International Conference on Software Testing, Verification and Validation, 2010.
- CHILLAREGE, R., BHANDARI, I., CHAAR, J., HALLIDAY, M., MOEBUS, RAY, B., WONG, M., "Orthogonal Defect Classification- A concept for In-Process Measurements", In *IEEE Transactions on Software Engineering*, 1992
- CONBOY, K. (2009). *Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development*. *Information Systems Development*, vol 20, n. 3, pp. 329-354.

- CONNOLLY, D., KEENAN, F., MCCAFFERY, F., "Developing acceptance tests from existing documentation using annotations" In Conference of 2009 ICSE Workshop on Automation of Software Test, AST 2009, 2009
- CONRADI, R., MARJARA, A.S., HANTHO, Ø, FROTVEIT, T., AND SKÅTEVIK B., in "A study of inspections and testing at Ericsson, Norway", 2000
- CORDEIRO, J. V. B. M. (2007) "Sistema Toyota de Produção: Novo paradigma produtivo ou estratégia de operações" In XXVII Encontro Nacional de Engenharia de Produção, 2007.
- CRISPIN, L., GREGORY, J. (2009). "Agile Testing. A practical Guide for Testers and Agile Teams" ADILSON WESLEY 2009.
- CUGOLA, G., GHEZZI, C. (1998). "Software Processes: a Retrospective and a Path to the Future" Software Process: Improvement and Practice, John Wiley & Sons, Ltd., Volume 4, Issue 3 Pages 99–182
- CMMI (2013) http://www.cmmi.de/cmmi_v1.2/browser.html
- DAVIS, G. (2008). "Managing the Test Process", IEEE 2000, 2000.
- DEBNATB N., ZARZAN F. A., MANTANO G., RIESCO D. (2007) "Transformation of BPMN in Sub processes Based in SPEM Using QVT" IEE EIT 2007 Proceedings.
- DIAS NETO, A. C., TRAVASSOS, G. H. (2006) "Uma Infraestrutura Computacional para apoiar o planejamento e controle de testes de software" Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2006.
- DIAS NETO, A. C., NATALI, A.C.C., ROCHA, A. R., TRAVASSOS, G. H. (2006) "Caracterização do Estado da Prática das Atividades de Teste em um Cenário de Desenvolvimento de Software Brasileiro" V Simpósio Brasileiro de Qualidade de Software – SBQS 2006.
- DIAZ, M., SLIGO, J. (1997) "How Software Process Improvement Helped Motorola" IEEE SOFTWARE 1997.
- DIESTE, O. PADUA, A.G. (2007) "Developing Search Strategies for Detecting Relevant Experiments for Systematic Reviews" First International Symposium on Empirical Software Engineering and Measurement, ESEM-2007.
- DIJKSTRA, E.W., (1972). "The humble programmer", Communication of the ACM, 15(10): p. 859-866, 1972

- DILL, D., D., PEARSON, A., W. (1991) "The self-designing organization: structure, learning, and the management of technical professionals" Technology Management: the New International Language (p. 33), 1991.
- DOBSON, J., "Performance Testing on an Agile Project" In Agile Conference, 2007, 2007
- DYBA T., KITCHENHAM, B. A., JORGENSEN, M. (2005) Evidence-Based Software Engineering for Practitioners IEEE Software 2005
- DYBA° T., DINGSØYR T., HANSEN G. (2007) "Applying Systematic Reviews to Diverse Study Types: An Experience Report" In First International Symposium on Empirical Software Engineering and Measurement, IEEE 2007
- DYBA° T., DINGSØYR T., (2008) "Empirical studies of agile software development: A systematic review" Elsevier Science Direct, Information and Software Technology 50 (2008) 833–859
- EA, (2013) "Enterprise Architecture" In <http://www.sparxsystems.com.au/products/ea/>
- ENGSTRÖM, E., RUNESON P., (2012) "Test overlay in an emerging software product line – An industrial case study, Inform. Software Technologies" (2012), <http://dx.doi.org/10.1016/j.infsof.2012.04.009>
- FABRI, J.A. (2010) In "<http://enghariasoftware.wordpress.com/>".
- FAGAN, M. E. (1976) "Design and code inspections to reduce errors in program development" IBM SYSTEM JOURNAL - DESIGN AND CODE INSPECTION NUMBER 3 1976
- FOWLER, M. "Mocks aren't Stubs", In <http://martinfowler.com/articles/mocksArentStubs.html>, 2009.
- FUGGETTA, A., (2000), "Software Process: a roadmap", In: Proceedings of the Conference on the Future of Software Engineering – International Conference on Software Engineering, pp. 25-34, Limerick, Ireland.
- FURTADO F., MORAIS E., PINTO A., MACIEL T. (2008) "Uso do EPF Composer para Adaptações de Processo de Desenvolvimento de Software", In Engenharia de Software Magazine, Edição 17, Ano 2, 2008. Centro de Informática da Universidade Federal de Pernambuco (UFPE), C.E.S.A.R – Centro de Estudos e Sistemas Avançados do Recife, Universidade Federal Rural de Pernambuco (UFRPE)

- GALLARDO-VALENCIA, R.E., SIM, S.E., "Continuous and Collaborative Validation-A Field Study of Requirements", In Conference of 2009 2nd International Workshop on Managing Requirements Knowledge, MARK, 2009
- GELPERIN, D., HETZEL, B. (1988) "The Growth of Software Testing" In Communications of the ACM 31 (June 1988), p. 687-95.
- GHINATO, P. (1995). "Sistema Toyota de produção: mais do que simplesmente Just-in-Time." Prod., São Paulo, v. 5, n. 2, Dec.
- GILB, T. (1985) "Evolutionary Delivery versus the Waterfall Model" In ACM SIGSOFT Software Engineering Notes Vol. 10 No 3 Jul 1985.
- GITTENS M., KIM Y., GODWIN D., In "The Vital Few versus the Trivial Many: Examining the Pareto Principle for Software", Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05)
- GOLDSMITH, R., GRAHAM, D., (2002) "The Forgotten Phase" In www.sdmagazine.com July, 2002.
- GOYAL, S., SCHILLER, J. (2007). "Major Seminar On Feature Driven Development Agile Techniques for Project Management and Software Engineering" Univ. Prof. Bernd Brügge, Technical University Munich.
- GRECHANIK M., XIE Q., FU C. (2009) "Experimental Assessment of Manual Versus Tool-Based Maintenance of GUI-Directed Test Scripts" In Proc. ICSM 2009, Edmonton, Canada, IEEE 2009
- GUNZ, H. P., PEARSON A. W., "Introduction of a matrix structure" R&D Management (June): 173-181. 1977.
- HASS, A. M. J., "Testing processes", In IEEE International Conference on Software Testing Verification and Validation Workshop ICSTW, 2008
- HANSEN, G. K., HAUGSET, B., "Automated Acceptance Testing Using Fit", In Proceedings of the 42nd Hawaii International Conference on System Science ,Hawaii, US, 2009
- HAUGSET, B. AND HANSEN, G.K., "Automated Acceptance Testing-A Literature Review and an Industrial Case Study" In Agile Conference 2008, 2008
- HERZLICH, P. (1993). "The Politics of Testing". Proceedings of 1st EuroSTAR conference, London, Oct. 25-28, 1993.

- HOLZ, H., KÖNNECKER, MAURER, F., 2001, "Task-Specific Knowledge management in a Process-Centred SEE", K.-D. Althoff, Feldmann, and W. Müller (Eds.): LSO 2001, LNCS 2176, pp. 163-177.
- IEEE 2008, Std 829™-2008 (Revision of IEEE Std 829-1998).
- ISO/IEC, 2003, The International Organization for Standardization and the International Electrotechnical Commission. ISO/IEC 15504:2008
- ISO/IEC, 2008a ISO/IEC 12207:2008 Systems and software engineering — Software life cycle processes The International Organization for Standardization and the International Electrotechnical Commission, Geneve: ISO, 2008
- ISO/IEC 2008b, "ISO 9001:2008 - Quality management systems - Requirement", The International Organization for the Standardization and the International E. Commission.
- JACOBSON, I. (1987) "Object Oriented Development in an Industrial Environment", OOPSLA '87 Proceedings, 1987.
- JIANG L., EBERLEIN A. (2009) "An Analysis of the History of Classical Software Development and Agile Development", Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA - October 2009
- JOHNSON, J. (2006) "Interview: Jim Johnson of the Standish Group" In <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>
- KEEN, P., NISSEN H., KLEIN H., Hirschheim R. (1991). "Relevance and rigor in information systems research: Improving quality, confidence cohesion and impact." in Information Systems Research:Contemporary Approaches & Emergent Traditions. North-Holland,Amsterdam.
- KELLY J. C., SHERIF J. S., HOPS J. (1992) "An Analysis of Defect Densities Found During Software Inspections" J. Systems Software, Feb. 1992, pp. 111-117.
- KESSLER, D., ANDERSEN, T. J., "Herding Cats: Managing Large Test Suites" In Agile Conference 2009, 2009
- KITCHENHAM, B. "Empirical Paradigm – The Role of Experiments" In Empirical Software Engineering Issues, LNCS 4336, pp. 25 – 32, 2007. Springer-Verlag Berlin Heidelberg 2007

- KLEIN, H., HIRSCHHEIMB, R. (2008). "The structure of the IS discipline reconsidered: Implications and reflections from a community of practice perspective", Elsevier, Information and Organization Volume 18, Issue 4, October 2008, Pages 280–302.
- KOSKELA, L. J., HOWELL, G. (2002). "The underlying theory of project management is obsolete", in: The PMI Research Conference, June 2002, Seattle, Washington.
- KRASNER, H., 2001, "Accumulating the Body of Evidence for The Payoff of Software Process Improvement". Software Process Improvement, IEEE, pp. 519–539
- LAITENBERGER, O., VEGAS, S., CIOLKOWOSKI, M. (2002) "The State of the Practice of Review and Inspection Technologies in Germany", Tech Report Number: ViSEK/011/E, 2002.
- LARMAN G., BASILI V.R, (2003), Iterative and Incremental Development: A Brief History In IEEE Computer, 36(6): 47-56, 2003
- MAFRA S., TRAVASSOS, G.H. (2006) "Estudos Primários e Secundários apoiando a busca por Evidência em Engenharia de Software", Relatório Técnico ES 687/06
- MARICK, B. (2000) "New Models for Test Development" in www.testing.com.
- MCGREGOR J. D., SYKES D. A., In "A practical guide to testing object-oriented software", Addison-Wesley, 2001.
- MELLO, M. (2011), "Melhoria de Processos de Software Multi-Modelos baseada nos modelos MPS e CMMI-DEV", Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, UFRJ, Março 2011.
- MONTONI, M. A. (2003) "Aquisição de Conhecimento: Uma Aplicação no Processo de Desenvolvimento de Software", apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, UFRJ, Agosto 2003.
- MPS.BR, Guia de Implementação, 2011.
- NONAKA, I., TAKEUCHI, H. (1997). "Criação do Conhecimento na Empresa." Rio de Janeiro: Campus, 1997.
- NONAKA, I., TAKEUCHI, H. (1986). "The new product development game", Harvard Business Rev., Jan./Feb., pp. 137-146, 1986
- OHNO, T. (1997). "O Sistema Toyota de Produção: além da produção em larga escala" In Bookman, Porto Alegre.

- OMG (2010). "Business Process Model and Notation (BPMN), Version 2.0 (2010)", Disponível em <<http://www.omg.org/spec/BPMN/2.0>>.
- OMG (2008) "Software Process Engineering MetaModel (SPEM) 2.0", April. 2008
- PAI, M., GORMAN, J.D. et al. (2004) "Systematic Reviews and meta-analyses: An illustrated, step-by-step guide", The National Medical Journal of India, vol. 17, n.2., 2004.
- POPPENDIECK, M., POPPENDIECK, T. (2003) "Lean Software Development: An Agile Toolkit", Addison Wesley 2003.
- PROWELL, S., TRAMMELL, C., LINGER, R., POORE, J. (1998) "Clean Room Software Engineering Technology and Process" The SEI Series in Software Engineering Institute, Addison Wesley 1998.
- PYHÄJÄRVI, M., RAUTIAINEN, K., ITKONEN, J., "Increasing understanding of the modern testing perspective in Software Product Development Projects" In System Sciences Proceedings of the 36th Annual Hawaii International Conference, 2003
- REIS, L. C. A. (2003) "Uma Abordagem Flexível para Execução de Processos de Software Evolutivos", Orientador Prof. Dr. Daltro José Nunes. Tese de doutorado, Universidade Federal do Rio Grande do Sul, Instituto de Informática, Porto Alegre, Brasil.
- RUNESON P., HÖST M. (2009) "Guidelines for conducting and reporting case study research in software engineering" In Springer Empirical Software Engineering (2009) 14:131–164 DOI 10.1007/s10664-008-9102-8.
- SAINT-EXUPERY, A.D., "Wind, sand and stars", London: Heinemann, 1954.
- SCHOTS N.C. L., SANTOS G., CERDEIRAL C., CABRAL, M.L., CABRAL R., SCHOTS M., NUNES E., ROCHA A. R. (2011) "Lições Aprendidas em Implementações de Melhoria de Processos em Organizações com Diferentes Características" in WAMPS 2011
- SCHULMEYER, G.G., MACKENZIE, G.R., Verification & Validation of Modern Software-Intensive Systems, New Jersey, Prentice-Hall Inc, 1999.
- SCHWABER, K., BEEDLE, M. (2002). "Agile Software Development with Scrum", Upper Saddle River, NJ, Prentice-Hall.
- SCHWABER K., SHUTERLAND (2012) "Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust." J., John Wiley & Sons, Inc., Hoboken, New Jersey.

- SHEWHART, W. (1939) "Statistical Method from the Viewpoint of Quality Control", Dover, 1986 (reprint from 1939).
- SHIMOKAWA, K., FUJIMOTO T. (2009). "O nascimento do Lean. Conversas com Taiichi Ohno e Eiji Toyoda" Bookman, 2009.
- SHINGO, S., (1986) "Zero Quality Control: Source Inspection and the Poka-Yoke System", ISBN 0915299313.
- SHULL, F., RUS, I., BASILI, V. (2000) "How Perspective-Based Reading Can Improve Requirements Inspections" IEEE July 2000.
- SOFTTEX, 2011. "MPS. BR: Melhoria de Processo do Software Brasileiro- Guia Geral (v.2011)". Disponível em: <http://www.softex.br/mpsbr>
- SOMMERVILLE, I "Software engineering", Addison Wesley, 8/e, 2006.
- STAPLETON, J. (1997) "DSDM Dynamic Systems Development Method: The Method in Practice" Addison-Wesley
- STOLBERG, S., "Enabling Agile Testing through Continuous Integration", In Agile Conference, 2009, 2009
- SVENSSON, H., HOST M., (2005). "Introducing agile process in a software maintenance and evolution organization" in Ninth European Conference on Software Maintenance and Reengineering (CSMR'05)
- THELIN T., RUNESON P., WOHLIN C. (2003) "An experimental comparison of usage-based and checklist-based reading" in IEEE Transactions on Software Engineering, vol. 29, no. 8, 2003.
- THOMKE, S. H. (1997) "The role of flexibility in the development of new products: An empirical study" Research Policy 26 (1997) 105-119 Elsevier Science.
- TRAVASSOS, G. H. , SANTOS, P. S. M, MIAN, P. , DIAS NETO, A.C. , BIOLCHINI, J., "An Environment to Support Large Scale Experimentation in Software Engineering" In IEEE International Conference on Engineering of Complex Computer Systems, Belfast Proceedings of ICECCS, 2008
- VARGAS, V. R., "Gerenciamento de projetos: estabelecendo diferenciais competitivos", BRASPORT 6ª Edição, 2006.
- VU, J.H., FROJD, N., SHENKEL-THEROLF, C., JANZEN, D.S., "Evaluating test-driven development in an industry-sponsored capstone project" In Conference of 6th International Conference on Information Technology: New Generations, ITNG, 2009

- WEINBERG G.M. (1971), "The psychology of computer programming", New York: Van Nostrand Reinhold, 1971
- WELLER, E. F. (1993), "Lessons from three years of inspection data" in IEEE Software September 1993
- WELLS, D. (2009) "Simplicity is the Key", <http://www.extremeprogramming.org/rules/simple.html> 2009
- WOOD, J., SILVER, D., "Joint application development", John Wiley and Sons, N.Y., 1989.
- YOUSSEF, M. (1992). "Agile manufacturing: a necessary condition for competing in global markets." Industrial Engineering. <http://www.accessmylibrary.com/article-1G1-13946099/agile-manufacturing-necessary-condition.html>
- ZUKAV, G. (2001). "The dancing Wu Li masters, An Overview of the new physics". HarperCollins Publishers (Perennial Classics Imprint)

APÊNDICE A – Métodos e Práticas Ágeis

Neste apêndice são apresentadas as práticas ágeis recuperadas a partir de uma revisão preliminar da literatura sobre métodos ágeis.

A.1 Empowered Teams, Open workspace

Jiang e Eberlein (2009) comentam a existência de trabalhos anteriores sobre a importância de motivar a equipe e, facilitar a interação entre seus membros delegando responsabilidades, em um ambiente de trabalho que favoreça a comunicação. E citam livro de Gerald Weinberg, publicado em 1971, sobre a psicologia e as relações de trabalho entre desenvolvedores de software, aonde é enfatizada a importância de alocação de um líder técnico na equipe capaz de tomar decisões e a importância de estimular uma interação face-a-face entre os programadores (WEINBERG, 1971) - também proposta pela prática *Open Workspace* do XP (BECK, 1993).

A prática ágil *Empowered Team* sugerida nas versões do DSDM (Atern e 4.2) recomenda o fortalecimento dos membros da equipe através de delegação de poderes para tomada de decisões, sem que seja necessário aguardar uma aprovação da gerencia superior (STAPLETON, 1997).

Ao trabalho de Weinberg é relacionado apenas a prática de *Empowered Team* do método DSDM, porém ao recuperar trabalhos sobre o método *Lean Development* verificamos que esta prática é bastante explorada também por esta proposta de desenvolvimento (POPPENDIECK & POPPENDIECK, 2003).

A.1.2 Times, Melhoria de Processo e Modelos de Produção

No *Lean Development*, a importância da autonomia e capacidade de decisão do desenvolvedor está relacionada à forma como a melhoria de processos deve ser alcançada: ao incentivar que as decisões sejam delegadas para o nível mais baixo (tático) em uma organização, objetiva-se desenvolver as capacidades da equipe em tomar decisões na melhoria das atividades que constituem o processo de trabalho (POPPENDIECK & POPPENDIECK, 2003).

Poppendieck (2003) apresenta os conceitos da metodologia *Lean* a partir de um breve resumo sobre a situação japonesa nos idos dos anos 40.

A Toyota, no início do processo de implantação de sua fábrica de automóveis se deparou com um grande desafio de produzir para poucos consumidores com baixo poder aquisitivo. Neste caso, o produto não poderia custar muito, e sendo a fatia de mercado pequena, a estratégia de produzir em grande escala para diminuição do

custo final não se tornava viável. O desafio passou a ser produzir uma quantidade pequena de carros, mas com o mesmo custo obtido em escala.

O Sistema Toyota de Produção (STP) surge a partir deste dilema, formando uma nova base para se pensar a fabricação, logística, e o desenvolvimento de produto. Taiichi Ohno engenheiro encarregado da oficina de produção foi o responsável pelas técnicas desenvolvidas a partir do princípio de eliminação do desperdício derivado do pensamento *lean*, como *just-in-time*, *kanban*, *autonomação* e controle da Qualidade Zero Defeitos (OHNO, 1997).

A *autonomação* é uma das bases do STP na qual é facultada ao operador ou à máquina, a autonomia de parar a produção sempre que for detectada qualquer anormalidade no processamento. *Just-in-time* descreve a possibilidade das partes adequadas à montagem de um produto em um fluxo de trabalho alcançar cada etapa da linha, no momento em que são necessárias, e somente na quantidade necessária. Uma empresa que estabelecesse esse fluxo integralmente poderia chegar ao estoque zero.

Ohno (1997) sinaliza que do ponto de vista da gestão da produção, a supressão definitiva de estoque seria um estado ideal. Sendo o automóvel um produto composto por milhares de componentes - gerado a partir da execução de um grande número de processos - seria muito difícil aplicar o *just-in-time* ao plano de produção de todas as etapas do processo de forma ordenada. Uma falha na previsão, um erro no preenchimento de formulários, produtos defeituosos e retrabalho, além de problemas com o equipamento e eventuais absenteísmos demandam a organização trabalhar com um estoque mínimo, pois um problema não tratado no início do processo, ou ainda nas etapas intermediárias sempre resulta em um produto defeituoso no final do processo.

O significado da palavra *kanban* em japonês é cartão de sinal. Este cartão é utilizado por uma determinada equipe responsável por uma determinada etapa de um processo produtivo, para sinalizar à equipe da etapa anterior que eles podem trabalhar, pois existe disponibilidade de mão-de-obra para tratar o que eles irão gerar de insumo (OHNO, 1997).

No entanto, apesar da notoriedade que algumas destas técnicas alcançaram não se deve atribuir seu sucesso apenas a sua aplicação isolada, e sim a uma estratégia contemplando princípios, métodos e técnicas de forma sistemática e concatenada.

Para que a implementação do *kanban* seja exitosa é necessário assegurar qualidade aos produtos intermediários em cada etapa do processo produtivo, isto evita que produtos defeituosos de uma etapa posterior não sejam enviados para a etapa

seguinte, ocasionando uma pane no *kanban* e no *just-in-time*, pois em uma situação em que o material chega com defeito e não existe um estoque mínimo para reposição, a linha de produção para (GHINATO, 1995).

O Controle de Qualidade Zero Defeitos (CQZD) é composto dos seguintes elementos: (a) TQC (Controle da Qualidade Total) – Sistema que integra o desenvolvimento da qualidade, a manutenção da qualidade e os esforços realizados por todos trabalhadores do setor produtivo na melhoria da qualidade dos produtos (b) CCQ (Circulo de Controle de Qualidade) constituído por um conjunto de colaboradores voluntários que se reúnem regularmente em busca de novas técnicas e abordagens para aumentar a qualidade no processo produtivo com autonomia para implementar as mudanças necessárias, porém subordinados a uma hierarquia, com sua conduta sintonizada com as diretrizes da alta administração. (c) Utilização de inspeção na fonte com o objetivo de determinar a relação causa-e-efeito entre erros cometidos e defeitos. Normalmente os defeitos são o efeito da utilização incorreta de algum (ou vários) dos fatores de produção. (d) Aplicação de dispositivos à prova de falhas (“*Poka-Yoke*”) (GHINATO, 1995).

Para que seja possível realizar inspeção em 100% dos casos, ao invés de utilização de um critério por amostragem se torna necessário a adoção de dispositivos que permitam que esta inspeção seja executada de forma rápida, liberando o profissional envolvido para focar sua atenção em outros aspectos das atividades em que está envolvido.

A partir da constatação desta necessidade, e da classificação dos tipos de erros na indústria de manufatura que podem gerar defeitos no produto, Shigeo Shingo (1986) desenvolveu um conjunto de cento e doze dispositivos para detectar defeitos na linha de produção com um custo mínimo e diminuindo o esforço humano para realizar esta verificação, conhecidos como *poka-yoke*. São vários os exemplos de *poka-yoke*: um gabarito que rejeita uma peça processada incorretamente, um *checklist* contendo procedimentos a serem realizados antes de ligar um equipamento, uma balança capaz de verificar o peso necessário que uma determinada peça e outros mais. Os *poka-yoke* podem ser utilizados no modo controle ou no modo advertência. Quando é utilizado controle, ao ser ativado a máquina e/ou linha de produção para automaticamente, até que o problema seja corrigido. No modo de advertência, quando ativado é acionado um alarme para avisar o trabalhador da detecção de um defeito. Em ambos os modos, sempre que um defeito é encontrado é realizada uma avaliação da causa-raiz do problema, para que problemas da mesma causa-raiz não venham a ocorrer.

Ao listar os benefícios na adoção de técnicas de leitura propostas para serem utilizadas em atividades de inspeção de requisitos em engenharia de software (como *checklist* e *scenario-based reading*) são percebidas algumas semelhanças, em relação ao uso dos dispositivos *poka-yoke* na indústria da manufatura (SHINGO, 1986). Através da adoção destas técnicas nas atividades de inspeção é possível: (a) aumentar a relação custo x efetividade do trabalho dos pesquisadores (b) reduzir a influencia dos inspetores nos resultados finais obtidos (c) prover um modelo para escrever documentos com maior qualidade (no caso da manufatura os modelos seriam os gabaritos utilizados nas inspeções) (SHULL *et al*, 2000).

O método FDD (*Feature Driven Development*) utiliza-se de inspeções na etapa de *design*, quando os diagramas e as classes já foram definidos e o código responsável pela implementação da funcionalidade será desenvolvido. Entretanto, não explicita a técnica a ser utilizada e formas para obtenção de um bom desempenho neste processo (GOYAL e SCHILLER, 2007).

As premissas do desenvolvimento *Lean* (*Lean Development*) derivados para uso em Engenharia de Software, em relação à melhoria dos processos são: (1) Uma organização madura olha para todo o sistema, e ao tomar iniciativas para aperfeiçoar o processo, considera o todo e não se concentra apenas em aperfeiçoar partes em separado (2) Uma organização madura estimula o aprendizado efetivo e prepara as pessoas que fazem o trabalho para que estejam aptas a tomar decisões.

A proposta do *Lean Development* ancora estas premissas aos conceitos de *autonomia* (*jidoka*) em que os desenvolvedores são incentivados a participar de forma mais ativa nos processos de melhoria de qualidade (POPPENDIECK & POPPENDIECK, 2003).

Esta abordagem sugere a formação de comunidades de especialistas para que boas práticas sejam reutilizadas e que um grupo de profissionais possa exercer consultoria nos projetos, sugerindo a adoção de uma estrutura matricial. Contudo não foi encontrada alguma referência sobre o armazenamento destas práticas em uma base de conhecimento, bem como a definição e manutenção das varias especializações que um processo de desenvolvimento de software demanda.

Como o conhecimento dos processos de trabalho de uma organização constitui o capital intelectual da organização, ele deve ser gerenciado de forma eficiente para garantir a sua preservação e permitir a sua constante evolução (DAVENPORT e PRUSAK, 1998) (MONTONI, 2003).

Esta necessidade de explicitar o conhecimento é enfatizada em artigo sobre o STP e as dificuldades de empresas ocidentais - especialmente brasileiras - em alcançar resultados similares às empresas japonesas, usando para isso as mesmas

abordagens de produção na indústria de manufatura, sendo mencionada a capacidade de troca de conhecimento tácito entre os orientais, em contraste com as culturas ocidentais que teoricamente demandam uma externalização do conhecimento (NONAKA e TAKEUCHI, 1997), (CORDEIRO, 2007).

A aquisição de conhecimento sobre melhorias a serem inseridas em um processo pode ser realizada através de uma avaliação dos resultados obtidos nas fases finais de execução de cada projeto (*post-mortem*), quando um grupo mantido pela organização com foco na manutenção e evolução dos processos de software (SEPG – *Software Engineering Process Group*) se reúne com os membros da equipe do projeto para avaliar os produtos, os tempos de execução, recursos utilizados e resultados obtidos para determinar eventuais melhorias nas atividades relacionadas.

Esta forma de avaliação está aderente aos princípios do CMMI e MPS.BR descritos nas duas premissas abaixo:

(1) Um sistema é mais bem gerido se subdividido em produtos de trabalho identificáveis, os quais são transformados a partir de uma entrada para um estado de saída, com o objetivo de atingir metas específicas,

(2) Uma organização madura é aquela em que as tarefas são planejadas e controladas para atendimento de um plano (SOFTEX, 2011), (CMMI, 2013).

A abordagem sugerida pelo Lean Development contrapõe a um modelo baseado na ideia de um produto sendo transformado gradativamente, e evoluído nas etapas do ciclo de desenvolvimento, a outro modelo com ênfase no fluxo das atividades.

Segundo Starr (1966) qualquer processo de produção pode ser visto como um sistema de entradas e saídas. Existe um conjunto de recursos que chamamos de entradas, e um processo de transformação opera sobre este conjunto, liberando-os em uma forma modificada que chamamos de saídas. O que usualmente é denominado gestão da produção é a gestão deste processo de transformação.

O controle da execução deste processo correspondente ao modelo de controle de gerência (*thermostat model*): (a) Existe um padrão de desempenho (b) O desempenho é medido na saída (ou entrada) (c) A possível variação entre o padrão e o resultado medido é utilizado para corrigir o processo e desta forma o padrão pode ser atingido (HOFSTEDE, 1978).

A principal diferença entre o ponto de vista de transformação e o ponto de vista do fluxo é que este último inclui o tempo como um atributo de produção. Como o tempo de entrega é afetado pela interdependência entre as tarefas, e a variabilidade de processamento das várias etapas do processo produtivo, o foco é direcionado à esta variabilidade e aos relacionamentos entre as tarefas - o que, segundo Koskela e

Howel (2003) não é facilmente visualizado na abordagem baseada em transformação.

Em relação aos objetivos do gerenciamento de projetos, a visão de fluxo privilegia a discussão sobre "o trabalho desnecessário que não deverá ser feito". O objetivo principal seria eliminar resíduos do próprio fluxo do processo. Tais princípios como a redução do tempo de atravessamento (*lead-time*)⁴, vantagem do tempo e redução de variabilidade são colocados em primeiro plano. Assim, a prescrição de gestão é completamente diferente em comparação com o ponto de vista de transformação. A primeira sugere reduzir a incerteza, enquanto o segundo aceita a incerteza existente (KOSKELA E HOWELL, 2003).

Já na abordagem de geração de valor, o ponto de vista está associado ao alcance do melhor valor possível do ponto de vista do cliente. A principal diferença entre o ponto de vista da transformação e o da geração de valor é que neste último, o cliente está incluído desde a concepção até a finalização do produto. Ao utilizar um modelo de transformação a premissa é que os requisitos existem no início, e podem ser decompostos durante as etapas de produção. O modelo baseado em valor admite que no início, as necessidades dos clientes não estão necessariamente disponíveis ou bem compreendidas, e que a apreensão destes requisitos pelas diferentes partes do projeto é um problema que se coloca.

Koskela e Howell (2003) argumentam que por vezes, estas três perspectivas são apresentadas como excludentes, e sinalizam a necessidade de integra-las, e conceber uma teoria de produção em que sejam considerados as transformações, o fluxo de trabalho e o conceito de valor.

Uma implementação de um modelo baseado em fluxo é o método *Kanban* proposto por Anderson para ser utilizado na melhoria de processos de desenvolvimento de software (ANDERSON, 2010).

O *Kanban* (com K maiúsculo para diferenciar do kanban proposto por OHNO, 1997) foi projetado por Anderson, ao modelar as atividades que fazem parte do ciclo de vida de desenvolvimento de software como uma cadeia de valor e, através de um sistema de acompanhamento e visualização, apoiar o monitoramento das entradas e saídas dos artefatos gerados em cada etapa do processo, com o objetivo de detectar gargalos no fluxo do trabalho.

⁴ Tempo de atravessamento em produção é a medida do tempo gasto por um sistema para transformar matérias-primas em produtos acabados disponíveis para uso (TUBINO, 1990; CORREA *et al.*, 2001).

Uma característica de um sistema puxado é que este limita a capacidade de trabalho em progresso (*Wip Work-In-Progress*) para uma quantidade acordada previamente. Como consequência desta limitação, apenas os trabalhos realizados na etapa em que existe o gargalo é que podem eventualmente ficar sobrecarregados. Como decorrência desta estratégia, em alguns momentos, alguns membros da equipe podem ter um momento de folga.

Com a determinação dos pontos do processo produtivo em que a capacidade de entrega está reduzida é possível realizar eventuais acertos na distribuição da equipe pelas etapas produtivas, e com isto obter uma cadência mais ou menos constante, apoiando uma melhor previsibilidade nos prazos e orçamentos.

Contudo a indústria de manufatura e indústria de software apresentam algumas características distintas. Na implementação de uma colaboração multifuncional entre participantes de etapas do ciclo de vida diferentes no desenvolvimento de software, deverão ser consideradas as demandas de especialização dos profissionais para a realização de algumas atividades do processo produtivo do *software*.

A.2 Integração Contínua (XP) e Entregas Frequentes (DSDM)

De acordo com Jiang e Eberlein (2009), as práticas de integração contínua e entregas frequentes de software aparecem em artigos no início dos anos 80 com o objetivo de antecipar problemas relacionados à falta de entendimento dos requisitos, e aumentar a sinergia com o cliente final.

Em um trabalho publicado por Gilb (1985) é proposto um modelo alternativo ao modelo cascata denominado "*Evolutionary Delivery*". Nesta proposta já é claramente defendida uma estratégia de planejamento das entregas a ser realizada conjuntamente com o cliente (jogos de planejamento), para que seja levado em conta o esforço de desenvolvimento versus o valor agregado ao negócio (*user-value to development-cost ratio for earliest implementation*), a utilização de inspeções e, o acompanhamento do usuário - tendo este um papel explícito de fornecedor de requisitos no processo de software.

Entretanto, apesar do ciclo de vida do processo proposto por Gilb possuir uma etapa específica para realização de testes de regressão, a partir do atendimento de mudanças solicitadas pelo cliente, alguns recursos ainda não estavam disponíveis como: (a) a possibilidade de automatização de *Build*, (b) notificação dos resultados de compilação, (c) publicação do novo código em área de homologação, e outros recursos que foram incorporados gradativamente (STOLBERG, 2009).

A.3 Iterações e Incrementos

Segundo Larman e Basili (2003) o modelo iterativo teve suas origens numa proposta de melhoria de qualidade, iniciada nos anos 30, baseada em um ciclo de *feedback-loop* denominado PDSA (*Plan-Do-Study-Act*) desenvolvido nos laboratórios BELL (SHEWHART, 1939).

O modelo teria sido utilizado pela primeira vez no projeto de desenvolvimento de um jato supersônico nos anos 50, sendo sua adoção considerada uma das causas mais relevantes para o sucesso desta empreitada.

Neste trabalho são descritos a evolução do método iterativo citando os modelos derivados *CLEANROOM* (PROWELL, 1998), *Spiral* (BOEHM, 1985), *Ada* (BOOCH, 1983), *Evolutionary Project Manager* (GILB, 1985) e o Trident/IBM. Sendo este último, o resultado de um projeto desenvolvido por uma divisão da IBM realizado em 1972 para desenvolver o sistema de comando e controle do submarino Trident da marinha americana.

A adoção de ciclos curtos, time-boxes com desenvolvimento iterativo são propostas nos métodos XP, DSDM e SCRUM.

A.4 Sprint Backlog & Product Backlog

Associada à proposta do Scrum (SCHWABER, 2002) de fatiamento do conjunto de funcionalidades em entregas a serem desenvolvidas em ciclos curtos e, controladas a partir de um *product backlog*, Jiang e Eberlein (2009) relacionam o método ASP (*Agile Software Process*) proposto por Aoyama (AOYAMA, 1998).

O objetivo deste método é fornecer um processo com capacidade de adaptação às mudanças do ambiente, no produto e mesmo no próprio processo, de forma rápida e flexível.

O processo constituído foi evoluído durante dez anos em projetos de software na Fujitsu, e tendo como domínio de problema softwares de comunicação em larga-escala. A ideia central consiste em desenvolver múltiplas funções de forma concorrente usando vários times de desenvolvimento geograficamente distribuídos. As funcionalidades entram em uma fila priorizada no backlog de produto.

Aoyama (1998) assinala os seguintes fatores de sucesso para sua implementação:

- (a) É preciso que já exista um processo de software definido para ser utilizado como base, desde que a proposta é modelar um conjunto de atividades já definidas de forma que sua execução possa ocorrer de forma incremental, *time-boxed* e concorrente. Aoyama sugere

que preferencialmente a organização apresente um nível de maturidade em processo análoga ao nível 3 CMMI (CMMI, 2013).

(b) É importante constituir um grupo de processos de engenharia de software. As funções deste grupo são extensas e incluem: melhoria do processo, transferência de tecnologia, e a gerencia das informações do Processo (PRIME – *Process Information Manager*) constituída das seguintes atividades:

- Apoio aos membros da equipe na execução das atividades individuais em uma linguagem visual utilizada para descrever o processo de desenvolvimento YPL (YAC – *Oriented Process Description Language*)
- Suporte ao planejamento e a execução do processo de software
- Coleta das estatísticas
- Consulta e Visualização das estatísticas
- Controle da segurança.

Porém para que os esforços deste grupo sejam compensatórios é fundamental não deixar de fora o ponto de vista do desenvolvedor.

(c) Foram decorridos dez anos até o processo se mostrar satisfatório, durante esta jornada dois pontos se mostraram relevantes:

- Evoluir um processo pode custar meses e até anos.
- É fundamental definir um roteiro (*roadmap*) da evolução pretendida para o processo.

(d) O PSEE (*Process Software Engineering Environment*) a ser utilizado deve ser capaz de coletar e disseminar informações da execução do processo nas formas estruturadas, semiestruturadas, e mesmo não estruturadas. Ele não deve ser inflexível e rigoroso a ponto de gerar perda de produtividade. O suporte fornecido por este ambiente deve ser em nível da atividade individual (*people-centered*).

(e) Ao aplicar o ASP em equipes de 10 pessoas o processo evoluiu se tornando mais simples e leve.

A.5 Small Releases

Em livro publicado em 1988, Gilb expõe o ciclo evolutivo adotado pelo método *Evolutionary Delivery* (EVO) na qual o escopo do projeto é quebrado em incrementos.

A entrega do produto a partir de incrementos permite logo no início do projeto que o gestor receba um *feedback* se a tecnologia utilizada e a forma de trabalho está

efetivamente trazendo valor ao negócio do cliente e, se a equipe está compreendendo o que os clientes estão demandando, e se eles tem conhecimento do valor do que está sendo entregue.

No método XP (BECK, 1999) esta prática é denominada *small releases*, sendo uma das quatorze práticas principais (*Core Practices*) incluída na categoria “*Continuous Process rather than Batch*”.

A.6 Active user involvement & On-site customer

Em livro escrito por Christopher Alexander, em 1979, é proposta uma nova teoria da arquitetura baseada na configuração de padrões de projeto (*design patterns*), sendo destacada a importância da participação ativa do usuário e sua presença física na localidade onde o projeto está sendo conduzido (ALEXANDER, 1979).

Esta prática é prescrita pelos métodos DSDM e XP através dos termos *Active user involvement (DSDM)* e *On-Site Customer (XP)*.

A.7 Planning game, Daily scrums & Regular build schedule

A necessidade de evolução de um plano em resposta às mudanças de negócio ou nas especificações técnicas está presente em artigo de Takeuchi e Nonaka (1986) sobre o processo de desenvolvimento de novos produtos. A mesma ênfase à necessidade de replanejamento por conta da aceitação de mudanças é contemplada nas práticas *Planning Game*, *Daily Scrums* e *Regular Build Schedule*.

No estudo foi analisado o desenvolvimento de novos produtos em empresas multinacionais no Japão e nos Estados Unidos (Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox e Hewlett-Packard). Sendo o principal critério na escolha destes projetos, as características inovadoras que os produtos desenvolvidos apresentavam e, sua visibilidade dentro da organização.

Através da realização de entrevistas com os participantes foi constatado que seis características eram comuns na condução das atividades entre as organizações:

- 1- *Built-in Instability*
- 2- *Self-Organizing project teams*
- 3- *Overlapping development phases*
- 4- *MultiLearning*
- 5- *Subtle Control*
- 6- *Organizational transfer of learning*

Built-in Instability

De acordo com esta característica, a alta gerência fornece uma visão geral dos grandes alvos do projeto, invariavelmente demandando um alto nível de inventividade e criatividade, desafiando a equipe a buscar novos meios de atingir estes objetivos, e

delegando poderes para isto. Este ambiente de desafio constante, em que regras podem ser redefinidas durante o desenvolvimento do produto, traz certa instabilidade, e ao mesmo tempo mantem a equipe permanentemente atenta sobre a possibilidade de buscar novas formas de trabalho.

Self-Organizing Project Teams

As equipes atuam com alto grau de autonomia, tomam iniciativas e aceitam riscos, desenvolvendo uma agenda independente. No dia-a-dia da gestão, a alta gerência raramente intervém, e a equipe é livre para definir a sua própria direção.

Além da autonomia, as equipes avaliadas apresentavam um alto compromisso com a busca por transcendência dos seus limites. Ao perseguir o que parecem à primeira vista objetivos contraditórios, eles inventam maneiras de ultrapassar limites até então intransponíveis na forma de pensar do status-quo da organização.

A diversidade obtida a partir do agrupamento de profissionais provenientes de setores diversos como marketing, sistemas, vendas e produção promove o surgimento de novas idéias e conceitos (*Cross-Fertilization*).

Overlapping Development Phases

Ao comparar as duas abordagens na execução de um projeto – sequencial/waterfall e sobreposição entre fases - Takeuchi e Nonaka (1986) comentam que apesar da primeira abordagem favorecer a minimização de riscos ao projeto, por terem marcos bem-definido e pontos de controle antes do início e fim de cada atividade, quando ocorre um gargalo em uma determinada fase, a execução do processo pode se tornar vagarosa, ou mesmo ser necessário suspender temporariamente a execução do processo.

Segundo os autores, uma vantagem de um processo que suporta sobreposição entre as fases é a responsabilidade pelo sucesso do projeto ser compartilhada pelas equipes, estimulando a cooperação e incentivando o foco na resolução de problemas, independente da fase em que o projeto se encontra.

Entretanto, para gerir um processo com estas características, um grande esforço gerencial é dispendido para: (a) manter uma boa comunicação entre os membros da equipe, (b) manter contato com fornecedores, (c) preparar os vários planos de contingência, e estar atento à possibilidade da ocorrência de conflitos no grupo.

MultiLearning

Como cada membro da equipe de projeto tem contato com vários participantes de etapas distintas do processo produtivo, o conhecimento dos vários aspectos de cada atividade é compartilhado, facilitando o entendimento dos pontos que podem ser

afetados na ocorrência de mudanças no segmento de mercado no qual estão inseridos.

Subtle Control

Apesar das equipes de projetos das organizações estudadas serem auto-gerenciáveis, em todos os projetos avaliados pelos autores foi necessário exercer um monitoramento para assegurar a realização de reuniões de ponto de controle do projeto, evitar desentendimentos e excesso de tensões, sendo este controle realizado de forma sutil.

Através do acompanhamento dos profissionais por seus pares, da seleção de participantes com perfis distintos entre si (nível de experiência, personalidades com características diversas) e apresentando um conjunto comum de crenças e valores.

Transfer of Learning

Em cada projeto avaliado de cada organização estudada, além de ter ocorrido transferência de conhecimento entre colaboradores do mesmo projeto, também foi observada propagação do conhecimento para outros setores da empresa. Esta propagação foi realizada de duas formas: (a) com a participação dos membros de um determinado projeto em outros projetos da organização (b) através da institucionalização de práticas e técnicas bem-sucedidas que foram desenvolvidas no âmbito dos projetos.

No resumo da avaliação realizada nas organizações, Takeuchi e Nonaka (1986) relacionam os seguintes fatores críticos para favorecer flexibilidade e velocidade no desenvolvimento de novos produtos: (a) adoção de um estilo de gestão capaz de acompanhar o desenvolvimento de produtos cuja evolução raramente acontece de maneira linear e estática. (b) Um tipo diferente de aprendizagem é necessário. Na abordagem proposta não somente especialistas serão responsáveis pelo desenvolvimento do produto. Em certos casos, os envolvidos serão incentivados a adquirir o conhecimento necessário e as habilidades durante a execução do projeto. (c) Algumas administrações consideram novos projetos como uma fonte geradora de futuros fluxos de receita, quando segundo os autores deveriam ser utilizados principalmente como agentes de mudança organizacional.

A.8 Self-managed teams; Empowering teams; Feature teams

Em artigo publicado em 1991 Dill e Pearson descrevem a importância de formação de equipes com capacidade de auto-organização em empresas cuja indústria apresenta um uso intensivo de tecnologia, com demandas frequentes por colocação rápida no mercado de produtos (*Time-To-Market*) contemplando uma nova mudança tecnológica caracterizada por um forte diferencial competitivo.

Nestas organizações, as propostas tradicionais para estruturar as equipes parecem não ser capazes de apoiar as mudanças no fluxo de trabalho resultantes das mudanças tecnológicas, como é o caso das estruturas organizacionais baseadas em projetos, em funções ou matriciais.

A adoção de uma estratégia matricial é questionada por alguns autores (GUNZ & PEARSON, 1977) considerando a sobreposição de atividades de planejamento, definição de orçamento e controle e recrutamento, quando alguns profissionais precisam se reportar a mais de uma chefia durante a execução dos projetos.

Com o aumento da competitividade, reflexo da globalização dos mercados, as pressões por obter um tempo de atravessamento menor demandam por novas capacidades das organizações, para responder a bruscas mudanças tecnológicas.

Para que a organização esteja apta a atender estas demandas é fundamental e o estabelecimento de um meio capaz de favorecer a comunicação entre as áreas funcionais. Este meio – chamado pelos autores de *interface* - pode ser constituído a partir da criação de um comitê, de uma força-tarefa, ou de times cujos membros são capazes de definir a melhor forma de organização que se adeque às necessidades do projeto a ser encaminhado (*Self-Organizing Teams*).

Os autores ressaltam algumas das vantagens de poder contar com profissionais com este perfil, a saber: (a) o estabelecimento de uma visão mais estratégica e generalista entre seus participantes; (b) o compartilhamento da responsabilidade entre a equipe promovendo a cooperação; (c) o incentivo ao aprendizado de resolução de problemas; também citados por Poppendieck & Poppendieck (2003).

A.9 Test-driven Development

Em um estudo comparativo em projetos de desenvolvimento de circuitos integrados foi constatado que tecnologias capazes de favorecer a realização de um projeto mais flexível também promoviam uma maior redução de custo. Sendo uma parcela significativa desta redução decorrente de uma maior tolerância a riscos - o que não acontecia quando a estratégia era pouco flexível demandando um dispêndio de recursos para mitigar a ocorrência de riscos relacionados à mudança de requisitos (THOMKE, 1997).

A flexibilidade de projeto é função de: (a) tecnologia escolhida para o desenvolvimento do projeto (b) escolhas dos projetistas (c) as características estruturais do *design*.

As empresas participantes do estudo que desenvolviam produtos com *designs* apresentando baixa flexibilidade eram obrigadas a investir mais em recursos em

pesquisa de mercado, para estar permanentemente a par das necessidades de cliente. Esta necessidade em realizar constantes pesquisas poderia ser mitigada se o projeto escolhido fosse capaz de se adaptar às mudanças solicitadas pelo cliente. Além disso, percebeu-se que o “congelamento” prematuro de um projeto nas etapas iniciais resultava em um produto com menos recursos e funcionalidades.

Foi realizado um estudo com um grupo de empresas que desenvolviam circuitos integrados (IC-*Integrated Circuits*) e que se utilizava de uma das duas tecnologias: sistemas baseados na tecnologia ASIC (*Application-Specific Integrated Circuits*) e sistemas baseados na tecnologia EPLD (*Electronically Programmed Logic Devices*).

Durante três meses foram conduzidas entrevistas com *designers* para obter entendimento dos desafios técnicos do desenvolvimento de ICs. Após estas entrevistas foi elaborado e enviado um questionário para 500 designers de ASIC e 500 designers de EPLD, escolhidos aleatoriamente deste ramo industrial nos Estados Unidos.

A média de solicitações de mudanças atendidas nos protótipos baseados na tecnologia EPLD foi 10 vezes superior ao atendimento das mudanças solicitadas pelos usuários em projetos ASIC. A média dos esforços HM gastos (horas trabalhadas ao mês por profissional) nas organizações em projetos de circuitos padrão EPLD foi metade da média dos mesmos esforços dispendidos em empresa que se utilizavam do padrão ASIC.

A utilização intensiva de testes no desenvolvimento dos circuitos padrão EPLD foi apontada como um fator de sucesso na obtenção de *feedback*. Tão logo já existia uma versão preliminar do protótipo do circuito esta versão já era colocada em prova.

No âmbito deste estudo foi relatada a estratégia de desenvolvimento de uma das organizações que fizeram parte do estudo na qual a flexibilidade de projeto estava baseada em: (a) habilidade de desenvolver sistemas de informação que descrevem a interação entre a arquitetura do produto e os detalhes de projeto físico (b) capacidade de decidir rapidamente mudanças críticas (c) capacidade de executar iterações de projeto baseadas em teste.

Contudo, não foram fornecidos maiores detalhes e características de como este desenvolvimento orientado a testes era realizado no âmbito dos projetos.

A.10 Refactoring

Dijkstra (1972) comenta sobre algumas das linguagens de programação fornecendo um painel do desenvolvimento de linguagens de programação:

(a) disposição de *libraries* de sub-rotinas atendendo necessidades básicas de abstração a serem utilizadas nas etapas de design da aplicação,

(b) desenvolvimento da linguagem FORTRAN trazendo facilidades às atividades de codificação, porém com pouco suporte às atividades de projeto de programa,

(c) desenvolvimento da família de linguagens Lisp (*List Processing*) composta de estruturas de dados elementares a serem manipuladas a partir de funções matemáticas,

(d) desenvolvimento da linguagem ALGOL e disposição de um conjunto de facilidades no uso de *typed functions* e parâmetros facilitando a implementações de sub-rotinas, porém demandando mais disciplina no projeto de programas,

(e) desenvolvimento da linguagem PL/I contemplando uma variedade de comandos e recursos que por vezes dificultava a manutenção e o entendimento das rotinas desenvolvidas.

Como consequência da disseminação e uso destas linguagens, um conjunto de melhores práticas em programação foi sendo formado.

Este cenário seria a base para gradativamente as entregas de software apresentarem menos falhas (*Virtually Free of Bugs*), demandando para isto um esforço de programação menor. Registra que as três condições capazes de viabilizar este cenário já eram preenchidas na época em que o artigo foi escrito: (a) maior aceleração no aumento do custo de pessoal utilizado nas atividades de desenvolvimento e testes se comparados com o aumento do custo do hardware, trazendo pressão por efetividade. (b) demanda da indústria por projetos de sistemas de larga escala com maior confiabilidade. (c) viabilidade técnica.

Em seu artigo cita alguns exemplos de técnicas de programação que visam a melhoria contínua, com a realização de mudanças no código fonte, porém preservando o atendimento aos requisitos definidos anteriormente. De forma a tornar o programa mais claro e fácil de modificar como é proposto na prática *refactoring* do método XP (BECK, 1999).

A.11 Baselined requirements

Em um artigo sobre desenvolvimento de requisitos, Bell e Thayer (1976) relatam as dificuldades encontradas na análise de problemas relacionados às atividades de especificação de requisitos reportados nas etapas de *design* e construção de código para o sistema de Mísseis Balísticos do departamento de defesa dos Estados Unidos, e sugerem o uso de uma classificação para facilitar a comunicação entre os analistas de requisitos e os projetistas.

Também registram que no contexto dos projetos de desenvolvimento avaliados na pesquisa, os requisitos não eram relacionados a uma *baseline* e rastreáveis em documentos de projeto. Desenvolvedores tomavam decisões no decorrer do projeto para implementação de mudanças a partir de reuniões com os clientes, sem que algumas destas informações fossem registradas e formalizadas.

Em alguns projetos existia uma disciplina na documentação dos requisitos e uma tendência dos membros da equipe em serem mais cuidadosos à cerca do monitoramento e controle das mudanças. Sendo neste caso observada uma quantidade menor de inconsistências e detectadas na maioria das vezes no início do projeto. Todavia mesmo nestes casos se tornava necessário a utilização de uma classificação que auxiliasse na detecção das possíveis causas dos problemas.

Neste estudo foi observado o desenvolvimento dos requisitos em dois tipos de projetos com tamanho e domínio de problema diferente: (a) Um sistema de informação para apoiar a contratação de estudantes da UCLA pela indústria. (b) Um sistema de controle de mísseis balísticos.

Além dos documentos de requisitos, foi distribuído para a equipe de design de ambos os projetos um formulário para classificação dos eventuais problemas que dificultaram ou impediram a realização do design de algumas das funcionalidades elicitadas (RPR - *Requirements Problem Report*). Neste formulário foram dispostas sete categorias de discrepâncias para serem escolhidas pelos projetistas e a possibilidade de informar mais uma categoria que não estivesse na lista apresentada. As categorias eram: (a) Ambiguidade, (b) Informação estranha, (c) Fato incorreto, (d) Inconsistência com outros requisitos, (e) Omissão, (f) Erro de digitação (*typo*), (g) *Better design possible* (ainda na etapa de análise foram sugeridos elementos de projeto que podem ser melhorados pelos projetistas).

Na etapa de classificação foram consideradas apenas as inconsistências que comprovadamente eram defeitos. Esta distinção foi facilitada por diretriz seguida pelo processo de desenvolvimento adotado, através da qual cada inconsistência reportada era enviada para avaliação por um grupo de especialistas, responsável pela avaliação da pertinência de cada apontamento de inconsistência, e nos casos em que era confirmado o defeito uma ação corretiva era tomada.

Após a categorização foi constatada similaridade nas distribuições por tipo de defeitos de cada um dos projetos de software selecionados para o estudo, sendo a categoria ambiguidade a que apresentou maior percentual de ocorrência, seguida por omissão, *better design possible*, fato incorreto, inconsistência com outros requisitos e *typo*.

O princípio “*Requirements are baselined at high-level*” do método DSDM (STAPLETON, 1997) incentiva a formalização dos requisitos de alto nível para o release corrente via gerencia de configuração. Após esta formalização os requisitos ficam congelados para que a implementação seja realizada. As seleções de quais requisitos estarão nesta condição é acordada com os usuários na iteração de *business study*.

A.12 Stakeholder Collaboration

A proposta da técnica JAD (*Joint Design Application*) desenvolvida pela IBM nos idos de 1970 é reunir os principais usuários, os gerentes e analistas de sistemas, para que através de um conjunto de reuniões sucessivas sejam elicitados os requisitos para o desenvolvimento de um novo sistema de informações a ser desenvolvido ou ainda para a manutenção de um determinado sistema existente.

O custo e o tempo associados à coleta de dados, análise e definição de requisitos podem ser significativamente reduzidos, e a participação de todos os profissionais favorece o estabelecimento de diferentes perspectivas sobre o sistema desejado, muitas vezes gerando novas idéias e soluções. Como as partes interessadas estão representadas na equipe JAD, eventuais conflitos e discrepâncias podem ser identificados e resolvidos durante a fase de definição do problema.

Esta técnica pode também ser utilizada para realizar análise de viabilidade, análise de custo benefício, e análise de risco. E muitas vezes para as especificações de projeto, tais como diagramas de fluxo de dados, diagramas de entidade relacionamento e diagramas de fluxo do sistema são gerados durante a sessão JAD.

Ela é utilizada desde 1970 em laboratórios de pesquisa da IBM (WOOD e SILVER, 1989).

A.13 Inspections

Inspeção de software é um tipo particular de revisão por pares em que o artefato de software é examinado por qualquer integrante da equipe do projeto, exceto o autor.

Revisão por Pares e Testes são métodos de verificação, que visam determinar se os produtos de uma determinada fase de um processo estão de acordo com os requisitos e condições definidas na fase atual e, nas fases anteriores do processo de desenvolvimento (SCHULMEYER e MACKENZIE, 1999).

Dois tipos de revisão se destacam: *Walkthrough* e Inspeções. As inspeções diferem do *Walkthrough* por seguirem um processo estabelecido. Fagan (1976) concebeu um processo constituído de uma sequencia de atividades que abrangem o planejamento, a apresentação dos artefatos, o estudo dos artefatos pelos inspetores, a reunião em que as discrepâncias apontadas são avaliadas se efetivamente são

defeitos ou se apenas não passam de falso-positivos, e da definição de continuidade da inspeção em busca de novos defeitos ou o término das atividades. Usando este processo ele realizou um experimento em que 67% dos defeitos detectados foram descobertos antes dos testes de unidade, utilizando-se inspeções de código. Os programas em que as inspeções foram feitas apresentaram menos 38% de falhas em comparação com aqueles em que a verificação não tinha sido executada a partir de um processo formal, e sim por Walkthrough de código, em que o programador apresenta o seu código e a documentação correspondente para a equipe de revisão.

Weller (1993) examinou três anos de inspeções realizadas na *Bull Information System*, incluindo informações sobre 11.557 defeitos e 14.677 páginas de documentação de projetos. Nestas avaliações percebeu que equipes de inspeção com melhor tempo de preparação obtinham melhores resultados, mesmo contendo menos participantes.

Sauer *et al* (2000) propuseram uma nova versão do processo tradicional de inspeção, através do qual a reunião passou a ser realizada de forma assíncrona e com a participação de equipes geograficamente distribuídas.

Estudos apontam para uma elevação substancial de custo de reparo à medida que o software evolui nas etapas de seu ciclo de vida (BOEHM, 1981) indicando que a detecção de inconsistências em etapas anteriores à codificação pode maximizar os resultados das inspeções.

Alguns métodos ágeis como o XP sugerem uma alternativa à adoção de inspeção de código através da prática *pair-programming* (BECK, 1999). Porém considerando que inspeção é uma atividade de verificação que pode ser aplicada em todos os artefatos de software, utiliza-la apenas nos códigos-fontes gerados pode postergar a detecção de defeitos para fases de projeto, em que haverá mais custos envolvidos na realização das correções e ajustes necessários.

A.14 Configuration Management

Boehm (1976) ressalta a importância de estabelecer um processo de gerência de configuração com o objetivo de controlar a evolução do software, e acompanhar o andamento das tarefas de desenvolvimento. Através do estabelecimento de *baselines* criadas a partir do agrupamento de itens de configuração, formados por um ou mais produtos derivados do trabalho da equipe de desenvolvimento do software.

Na criação da *baseline* é realizada uma revisão de projeto preliminar, quando é estabelecida uma gestão de configuração formal, em que as alterações propostas deverão ser aprovadas por representantes de todas as partes envolvidas.

No método DSDM (STAPLETON, 1997) a gerencia de configuração compõe uma das sete *Core Techniques*.

A.15 Integrated Testing

Sommerville (2006) registra a importância de estabelecer uma estratégia de teste através da qual deverá ser verificado se as diferentes partes do produto estão aptas a interagir de acordo com os requisitos funcionais e não funcionais elicitados: (a) Integração Bottom-Up: As rotinas e componentes de mais baixo nível são desenvolvidas e testadas de forma incremental e integradas de forma a compor a funcionalidade pretendida. (b) Integração Top-Down: O componente de mais alto nível é desenvolvido e testado, através da adoção de *stubs* ou *mocks*. Na sequência, as demais rotinas e/ou componentes chamados por este primeiro componente são desenvolvidos, integrados e testados.

No método DSDM (STAPLETON, 1997) é previsto uma etapa para a integração e testes dos componentes desenvolvidos a partir da execução dos vários ciclos de iteração.

A.16 Simple Design

O conceito de Design Simples para o XP está associado a quatro atributos relacionados à qualidade do software: (a) Testabilidade- À medida que o sistema é desenvolvido são criados testes de unidade e de aceitação, podendo ser executados a qualquer momento em esquema de regressão com o objetivo de detectar eventuais defeitos inseridos no software durante a evolução do produto. (b) Inteligibilidade – Está relacionada à facilidade que os membros da equipe conseguem acessar e entender trechos e rotinas de código. (c) Facilidade de compreensão- está associado à capacidade do conhecimento sobre o software ser rapidamente apreendido por novos membros da equipe. (d) Facilidade em explicar as rotinas e componentes – está associado à facilidade em demonstrar como os vários componentes que compõem o software trabalham em conjunto. Complexidade desnecessária deve ser retirada através da execução de re-fatorações de código, tantas quanto forem necessárias (WELLS, 2009).

Bourque *et al* (2002) ao examinar resultados de conferências sobre Engenharia de Software (IEEE, SEBOOK, ISESS) com o objetivo de elencar princípios fundamentais em SE relacionou uma lista de 65 candidatos incluindo entre eles a necessidade de minimizar as iterações entre os componentes, investir na compreensão do problema, projetar as rotinas pensando na possibilidade de reuso, associados ao conceito de *Design Simples*.

APÊNDICE B – Características de Agilidade

Neste apêndice são apresentadas as características de agilidade recuperadas a partir da execução de protocolo de quasi-revisão sistemática e posteriormente avaliadas quanto a pertinência e relevância a partir da execução de um survey (Abrantes, 2012).

Característica	Descrição
Adaptabilidade	Habilidade e capacidade de adaptar rapidamente o processo para atender e reagir a mudanças de última hora nos requisitos e/ou mudanças de ambiente, bem como atender e reagir a riscos ou situações não previstas.
Auto-organização	As equipes definem as melhores maneiras de se trabalhar; elas são autônomas e podem se auto organizar de acordo para completar os itens de trabalho.
Convergência	Atacar os riscos pro ativamente faz com que o sistema se torne mais perto da realidade procurada a cada iteração. Na medida em que esta ação progride, o sistema é entregue em incrementos. Tudo que estiver ao alcance dos envolvidos no desenvolvimento do software é feito para garantir o sucesso de modo mais rápido.
Emergência	Os processos, princípios e estruturas de trabalho são reconhecidos durante o processo de execução, não sendo definidos a priori; é permitido e aceito que tecnologias e requisitos emergjam durante o ciclo de vida do produto.
Equipes Locais	Para algumas metodologias isto significa equipes localizadas nas mesmas salas ou em salas adjacentes; isto funciona para equipes de 4 a 8 pessoas. Todas as metodologias são sensíveis à localização das equipes e estão fortemente baseadas em canais de comunicação ricos e rápidos, apoiando a redução de documentação a ser elaborada e mantida.
Equipes pequenas	Equipes pequenas, e pequeno número de equipes por projeto, são necessários para promover um ambiente colaborativo e requer menos planejamento para coordenar as atividades dos membros das equipes.
Incorporação de Realimentação (<i>feedback</i>)	As equipes devem ser capazes de receber e procurar continuamente por <i>feedback</i> de modo mais frequente e com mais rapidez.
Leanness	Esta característica está relacionada com a eliminação de perdas e com a habilidade de realizar mais trabalho com menos esforço; é uma característica de processos ágeis que requer o mínimo necessário de atividades para mitigar riscos e alcançar metas; todas as atividades que não são necessárias devem ser removidas do processo de desenvolvimento.
Modularidade	Esta característica permite que um processo seja particionado em componentes chamados de atividades, tornando viável adicioná-los ou removê-los de um processo

	quando necessário.
Orientação a Pessoas	Privilegiar pessoas em detrimento de processos e tecnologias; desenvolvedores são fortalecidos e encorajados a aumentar sua produtividade, qualidade e desempenho; comunicação e cooperação são fundamentais e necessárias; reuniões em pé e workshops de reflexão dão às pessoas a chance de expor suas preocupações.
Reflexão e Introspecção	Acontecem reuniões no final de cada subprojeto ou iteração, nas quais cada membro da equipe pode discutir o que está sendo bem feito e o que precisa ser melhorado.
Ser Colaborativo	Esta é uma atitude por parte dos membros da equipe de desenvolvimento, dentre os quais a comunicação é encorajada para disseminar informação e apoiar integração rápida de incrementos de software.
Ser Cooperativo	Interação aberta e proximidade entre todos os <i>stakeholders</i> (especialmente entre clientes e desenvolvedores); o cliente deve ser um elemento ativo no processo de desenvolvimento e deve prover realimentação (<i>feedback</i>) de modo regular e frequente.
Ser Incremental	Não tentar construir o sistema todo de uma só vez; o sistema deve ser particionado em incrementos (pequenas releases com novas funcionalidades) desenvolvidos em ciclos rápidos e paralelos; quando o incremento estiver completo e testado, ele é integrado ao sistema.
Ser Iterativo	Usar vários ciclos curtos guiados por <i>features</i> do produto, nos quais certo conjunto de atividades é completado em poucas semanas; estes ciclos são repetidos muitas vezes pra refinar as entregas.
Testes Constantes	Para prevenir a degradação da qualidade por causa de programas de <i>releases</i> curtos, um alto grau de ênfase é colocado no teste do produto ao longo de seu ciclo de vida; testes de integração devem ser automatizados com <i>builds</i> diárias bem como a execução de testes de regressão para garantir que todas as funcionalidades operem adequadamente.
<i>Time Boxing</i>	É o estabelecimento de limite ou fatias de tempo para cada iteração programada. Grandes esforços de desenvolvimento são divididos em múltiplas entregas desenvolvidas de modo incremental e concorrente, de maneira previsível.
Transparência	O método ou processo ágil deve ser fácil de aprender e de ser modificado, além de estar adequadamente documentado.

APÊNDICE C – Atividades do Processo de Teste Padrão

Neste apêndice são relacionadas as atividades que constituem o processo proposto por Dias e que será utilizado como processo de teste de padrão no estudo proposto (Dias Neto, 2006).

C.1 Atividades do Subprocesso de Planejamento de Testes.

1. **Planejar Testes** – Definir objetivo, escopo, a abordagem de teste, recursos físicos, RH e cronograma.
 - 1.1 Caracterizar testes (características a serem testadas e aquelas que não serão)
 - 1.2 Planejar estratégia para testar produto.
 - 1.3 Definir atividades de Teste. (atividades necessárias para preparar e executar os testes)
 - 1.4 Planejar recursos humanos
 - 1.5 Planejar recursos físicos
 - 1.6 Planejar tempo para testes
 - 1.7 Planejar Custos.
 - 1.8 Especificar Necessidade de Treinamento
 - 1.9 Identificar Riscos
2. **Projetar Testes** – identificar os procedimentos e CTs apresentado os critérios de aprovação.
 - 2.1 Selecionar itens de teste e características
 - 2.2 Descrever estratégia para o Projeto de Teste.
 - 2.3 Identificar Casos e Procedimentos de Teste.
 - 2.4 Priorizar Casos e Procedimentos de Testes
 - 2.5 Especificar Casos de Testes.
3. **Especificar Casos de Teste** - Definir os CTs incluindo dados de dados de entrada, resultados esperados, ações e condições gerais para execução.
 - 3.1 Caracterizar Caso de teste (qual requisito e característica associada)
 - 3.2 Especificar entrada e resultado esperado.
 - 3.3 Definir recursos para o Caso de Teste
 - 3.4 Especificar restrições de uso.
 - 3.5 Definir dependências entre Casos de Testes.
4. **Definir Procedimentos de Teste** - Descrever os passos para executar um conjunto de Casos de Teste.
 - 4.1 Definir objetivos de cada procedimento de teste, (quais itens de teste estão associados)
 - 4.2 Definir requisitos para a Execução do Procedimento.
 - 4.3 Descrever passos para a execução de cada procedimento.
 - 4.4 Construir Procedimento de Teste. (por meio de criação de scripts)

C.2 Atividades do Subprocesso de Execução dos Testes.

1. **Executar Testes** – Constituído das atividades de configuração do ambiente de testes, execução dos procedimentos e registro de incidentes durante a execução.

- 1.1 Configurar Ambiente de Teste.
- 1.2 Executar Procedimentos de Teste.
- 1.3 Registrar Incidentes de Teste.

2. **Analisar Resultados dos testes** – A análise dos resultados permite a organização avaliar o seu processo de testes, permitindo a proposição de melhorias em suas atividades.

- 2.1 Resumir Testes.
- 2.2 Registrar Dados dos Testes.

C.3 Atividades de Monitoramento, Controle e Replanejamento.

Além das atividades acima registradas deverão ser incluídas atividades de Monitoramento, controle e replanejamento para comparação do planejado versus realizado, em termos de esforço, recursos envolvidos, custos, tempo e cronograma. Como já existem atividades de planejamento e registro de incidentes de testes no processo original proposto por Dias Neto (2006) o monitoramento, controle e replanejamento poderá se utilizar das informações provenientes destas atividades para avaliação do desempenho do projeto de testes e, nos casos em que for necessário definir um replanejamento.

C.4 Atividade de Encerramento das atividades de Teste

Da mesma forma, apesar de não ter sido incluída uma atividade de encerramento dos testes, ao final do processo proposto originariamente os dados armazenados na subatividade "Registrar Dados de Testes" e aqueles derivados das atividades de planejamento fornecem subsídios para a realização do fechamento dos testes.

APÊNDICE D – Mapeamento Práticas Ágeis versus Atividades de Teste

Neste apêndice são descritas as práticas ágeis recuperadas a partir da execução de protocolo de quasi-revisão sistemática e posteriormente avaliadas quanto a pertinência e relevância a partir da execução de um survey, e também o mapeamento realizado entre estas práticas e Atividades de Testes, resultado de revisão por pares. As práticas ágeis e o relacionamento com as atividades foram extraídas do trabalho de Abrantes (2012).

D.1 Backlog de Produto.

Esta prática inclui tarefas para criação de uma lista de backlog de produto, e seu controle durante o processo de inserção, remoção, atualização e priorização dos itens da lista. A lista de backlog de produto define tudo o que é necessário para o produto final baseado no conhecimento atual que dele se tem. O backlog de produto define o trabalho a ser feito no projeto, incluindo uma priorização e constante atualização da lista de requisitos para o sistema sendo construído ou melhorado. Itens de backlog podem incluir, por exemplo, funcionalidades, correção de erros, defeitos, requisições de melhorias, atualizações de tecnologia, etc. Questões que requeiram solução antes de outros itens de backlog serem trabalhados, também podem estar na lista.

Atividade	Embasamento: o backlog de produto ..
Planejar Testes	pode apoiar a definição dos itens a serem testados. Pode também facilitar um acompanhamento para manter o plano de testes atualizado.

D.2 Cliente presente

Em termos práticos, isso significa colocar o cliente fisicamente próximo aos desenvolvedores ou mover os desenvolvedores para próximo do cliente. Esta prática indica que o cliente deve fazer parte da equipe de desenvolvimento. Para esclarecer e validar requisitos e estabelecer prioridades, um representante de cliente presente trabalha junto da equipe. Trabalhando junto dos desenvolvedores todo o tempo, o cliente pode responder perguntas, resolver questões, estabelecer prioridades, fazer testes de aceitação e assegurar que o desenvolvimento tenha o progresso esperado. Quando surgirem questões, os programadores podem resolver imediatamente com o cliente, ao invés de tentar imaginar quais seriam suas preferências. Esta prática também leva o cliente a mudar mais prontamente os requisitos, ajudando a equipe a mudar o foco dos esforços de desenvolvimento para as necessidades mais prementes.

Atividade	Embasamento: o cliente presente ...
Planejar Testes	pode auxiliar na solução de eventuais questões incidentes, bem como na priorização dos itens a serem testados e no estabelecimento de critérios para aceitação.
Projetar Testes	pode apoiar a identificação de casos de teste.

D.3 Design simples

A ênfase desta prática está em projetar a solução mais simples possível que seja aceitável no momento. Complexidade desnecessária e código extra devem ser removidos assim que reconhecidos. Não se devem incluir aspectos adicionais aos artefatos sem uma boa justificativa para tal. A prática do design simples requer que a equipe não projete para satisfazer necessidades futuras, as quais os desenvolvedores não devem tentar prever. A abordagem de desenvolvimento mais efetiva em termos de custo deve focar em resolver os problemas de hoje ao invés de projetar para mudanças futuras que não se sabe se realmente ocorrerão. Devesse fazer a coisa mais simples que possa funcionar.

Atividade	Embasamento: o design simples, sem complexidade desnecessária ...
Projetar Testes	pode facilitar a identificação de casos e procedimentos de teste.
Especificar Casos de Teste	pode facilitar a identificação de restrições e dependências com outros casos de teste.
Definir Procedimentos de Testes.	pode facilitar a identificação dos passos a serem seguidos durante os testes

D.4 Equipe completa

Refere-se à prática de incluir todos os perfis e perspectivas necessários na equipe para que ela possa ter bom desempenho, enfatizando o espírito de equipe, com todos os seus membros compartilhando um propósito e apoiando-se mutuamente. Clientes, usuários e demais interessados devem ter um envolvimento direto no projeto, a fim de possibilitar entender o comportamento do sistema mais cedo no ciclo de vida.

Atividade	Embasamento: ter uma equipe completa, envolvendo diferentes perfis, é benéfico para o processo de teste de software. Por exemplo, um especialista em segurança pode apoiar ...
Projetar Testes	na identificação de casos de teste envolvendo questões de autenticação, autorização e auditoria.
Especificar Casos de Teste	na especificação detalhada de casos de teste envolvendo questões de autenticação, autorização e auditoria.

D.5 Jogo de planejamento

Juntos desenvolvedores e clientes atuam no jogo de planejamento no qual o cliente escolhe as histórias de usuário que incluem os requisitos mais importantes a serem incluídos em uma entrega curta e incremental. Cada incremento curto implementado é experimentado pelo cliente. As histórias remanescentes são reavaliadas em termos de requisitos e prioridades, sendo o jogo de planejamento jogado novamente para definir o próximo incremento a ser implementado. A meta do jogo de planejamento é balancear os interesses do cliente com a capacidade da equipe. O planejamento é contínuo e progressivo. Os desenvolvedores estimam o custo das funcionalidades candidatas e o cliente as prioriza com base no custo e no valor agregado para o negócio. Uma das grandes vantagens do jogo de planejamento é a participação ativa do cliente e da equipe, com o processo de desenvolvimento sendo conhecido por todos. Diretrizes que levam a decisões relacionadas com liberações ou iterações específicas ficam claras para todos, pois cliente e equipe as definem juntos. Após as histórias de usuário terem sido definidas, a equipe de desenvolvimento fornece ao

cliente uma estimativa de tempo para implementar cada uma delas. O cliente então prioriza as histórias considerando estas estimativas. Posteriormente a equipe informa ao cliente o tempo que irão trabalhar no próximo incremento e baseado nisso o cliente seleciona as histórias que a equipe irá implementar em seguida. Os desenvolvedores então dividem as histórias em tarefas, mas sem envolver o cliente com detalhes de implementação.

Atividade	Embasamento: o jogo de planejamento, ...
Planejar Testes	sendo contínuo e progressivo, com prioridades estabelecidas pelo cliente, pode apoiar o estabelecimento de um plano de testes alinhado com as necessidades do projeto.
Projetar Testes	com prioridades estabelecidas pelo cliente, pode apoiar o estabelecimento de prioridades no projeto de teste.

D.6 Metáfora

Esta prática consiste em apresentar uma estória simples e compartilhada que explica a essência de como o sistema funciona para dar a desenvolvedores e cliente um entendimento comum do projeto. De certo modo, a metáfora serve como uma arquitetura de alto nível para o software. A metáfora serve para fazer a ligação de um domínio conhecido com um domínio com o qual não se está familiarizado. Pensando sobre uma metáfora apropriada, os desenvolvedores expandem suas perspectivas de análise da aplicação sendo desenvolvida. Há dois propósitos principais para a metáfora. Um é a comunicação. A metáfora preenche a lacuna entre desenvolvedores e usuários assegurando facilidades na discussão sobre o software e no fornecimento de exemplos. O segundo propósito é que a metáfora a equipe no desenvolvimento de uma arquitetura para o software.

Atividade	Embasamento: o entendimento comum do projeto pode ...
Planejar Testes	apoiar a busca de um planejamento adequado para os testes.
Projetar Testes	facilitar o projeto de teste, na identificação de casos e procedimentos de teste.

D.7 Liberações frequentes

Esta prática visa maximizar o retorno dos projetos assegurando que o maior valor de negócio possível seja entregue ao final de cada release e que cada release tenha uma duração curta. Isso é feito através do processo contínuo de priorização que seleciona sempre as histórias de maior valor para serem implementadas primeiro. Além disso, procura antecipar o retorno entregando software rapidamente. Ciclos curtos reduzem os riscos, possibilitando ao cliente terminar rapidamente com projetos que não agreguem valor para o negócio. Além disso, ciclos de Liberações frequentes ajudam a lidar com mudanças nos requisitos e reduzem o impacto de erros de planejamento. Ao final de cada release, o cliente revê todo o produto podendo identificar defeitos e fazer ajustes nos requisitos futuros.

Atividade	Embasamento: liberações frequentes ...
Executar Testes	influenciam as iterações ou quantidade de vezes que execuções de teste devem acontecer.
Analisar Resultados	influenciam as iterações ou quantidade de vezes que análise de resultados devem acontecer.
Monitorar e Controlar o Processo de Teste	influenciam as iterações ou quantidade de registros das tarefas executadas e dos resultados obtidos.

D.8 Reuniões diárias

As reuniões diárias em pé são reuniões rápidas, geralmente de 15 minutos, organizadas para acompanhar o progresso do projeto, destacar questões importantes e organizar as atividades diárias. Cada membro da equipe relata rapidamente no que está trabalhando e o progresso já alcançado. Durante a reunião todos devem ficar de pé, para encorajar os participantes a serem objetivos, não ultrapassar o tempo previsto para a reunião, além de manter todos alertas e com atenção voltada para os assuntos tratados.

Atividade	Embasamento: o acompanhamento do progresso do projeto pode ...
Planejar Testes	melhorar a comunicação e auxiliar a elaboração de um plano de teste alinhado com a realidade do projeto.
Projetar Testes	facilitar a integração das atividades de teste com as de desenvolvimento.
Especificar Casos de Teste	
Definir Procedimentos de Teste.	
Executar Testes	
Analisar Resultados	
Monitorar e Controlar o Processo de Teste	

D.9 Visibilidade de projeto

Projetos ágeis por sua natureza estão continuamente mudando (planos, modelos, código e demais artefatos). Deve haver esforços para fornecer às equipes o status do projeto em que estão engajadas. A psicologia mostra que quanto mais imediata a retroalimentação, mais rapidamente as pessoas mudam o comportamento para se adequar a novas situações. Pode ser criado um painel na web para manter a qualquer tempo o status e as métricas relacionadas com o progresso do projeto. Múltiplos painéis podem ser usados para disponibilizar diferentes tipos de informação que atendam a todos os níveis organizacionais necessários. O avanço do projeto com relação às histórias de usuário que as equipes se comprometeram a entregar no final das iterações deve ser incluído. Modelos devem se tornar acessíveis para todas as equipes.

Atividade	Embasamento :fornecer às equipes o status do projeto pode ...
Planejar Testes	melhorar a comunicação e auxiliar a elaboração de um plano de teste alinhado com a realidade do projeto.
Projetar Testes	facilitar a integração das atividades de teste com as de desenvolvimento.
Especificar Casos de Teste	
Definir Procedimentos de Teste.	
Executar Testes	
Analisar Resultados	
Monitorar e Controlar o Processo de Teste	

APÊNDICE E – Planilha p/ Comparação Processo de Teste Padrão e Processo de Teste da Organização

Neste apêndice são apresentados os resultados da comparação realizada entre as atividades do Processo de Teste da Organização e o processo de Testes adotado como padrão. As descrições de cada uma das atividades de Testes do processo padrão foram extraídas de Dias Neto (2006).

E.1 Classificação

Este documento apresenta a comparação entre as atividades de um Processo de Teste da organização selecionada para o estudo de caso e o Processo Padrão de Teste proposto por DIAS NETO e TRAVASSOS (2006), envolvendo os artefatos gerados por cada atividade, tarefas contidas em cada atividade e perfis envolvidos. Abaixo seguem os critérios de classificação utilizados neste mapeamento.

Classificação		Critério
EQU	Atividade equivalente	A Atividade do Processo de Teste implementado na organização é compatível com a definida no Processo Padrão.
EQUt<>	Atividade equivalente e realizada em etapa distinta	A Atividade do Processo de Teste implementado na organização é compatível com a definida no Processo Padrão, porém ocorre em etapa anterior ou posterior.
EQU-	Atividade equivalente com complementação	Na relação das tarefas, artefatos e perfis definidos na atividade do Processo de Teste da organização falta(m) um ou mais itens, presentes no Processo Padrão.
EQU-t<>	Atividade equivalente com complementação e realizada em etapa distinta	Na relação das tarefas, artefatos e perfis definidos na atividade do Processo de Teste da organização falta(m) um ou mais itens, presentes no Processo Padrão. Além disso, a atividade é realizada em etapa distinta da etapa em que é proposto no Processo Padrão.
EQU+	Atividade equivalente com elementos adicionais	Existe(m) tarefa(s) e/ou artefato(s) e/ou perfil(s) definido(s) na atividade do Processo de Teste da organização e que não está (ão) definido(s) na atividade associada no Processo Padrão.
NEQ	Não Equivalente	Não existe atividade no Processo de Teste da organização equivalente a esta atividade do Processo Padrão
INE	Inexistente	Não existe atividade no Processo Padrão equivalente a esta atividade do Processo de Teste da organização

E.2 Atividades

1. Macro-Atividade Planejar Testes (Processo Padrão)

1				
Macro-Atividade de Teste Processo Padrão		Macro-Atividade de Teste do Processo da Organização		Classificação
Planejar Testes		Planejar Projeto		EQU-/EQU+
Descrição (Planejar Testes - Processo Padrão):		Descrição (Planejar Projeto – Processo da Organização):		
Nesta atividade o Gerente de Teste deverá estimar os custos, definir o cronograma das atividades de teste, recursos envolvidos, e definir os itens a serem testados, as estratégias, métodos e técnicas de teste a serem adotadas.		As atividades de Teste são planejadas conjuntamente com as atividades de Planejamento do Projeto em uma única atividade denominada "Planejar Projeto". Nesta etapa, o Gerente do Projeto define o Plano do Processo para o projeto, incluindo as atividades dos diversos processos que compõem o desenvolvimento do produto, como Processo de Teste, Gerencia de Configuração, Gerencia de Requisitos, Monitoramento e Controle e outros. Para cada um destes processos são identificadas as atividades a serem executadas, o escopo, as características, o ciclo de vida do projeto e, as alterações propostas em relação ao processo padrão. É gerada a EAP (Estrutura Analítica do Projeto), o Plano de Custos, Plano de Riscos, Plano de Comunicação e Dados, Plano de Medição, Plano de Treinamento e o Plano de Gerencia de Configuração.		
Artefatos de entrada:	Template Plano de Teste Processo Padrão (Dias Neto e Travassos, 2006).	Artefatos de entrada:	Contagem Ponto de Funções, Repositório de Estimativas, Termo de Abertura do Projeto.	
Artefatos de saída:	Plano de Teste.	Artefatos de saída:	Plano do Projeto (incluindo atividades, tempos, recursos e riscos dos Testes).	
Atividade:		Atividade:		
1.1 Caracterizar Testes		<i>Processo da Organização não possui atividade correspondente nesta etapa do ciclo de vida do Teste.</i>		EQU-t<>

Descrição:		Descrição:	
Durante esta atividade, deverão ser caracterizados os testes a serem executados. Nesta etapa serão especificadas as seguintes informações: Itens de teste, Características do produto a serem testadas (como interface, desempenho, segurança ou funcionalidade) e características que não serão testadas. Devem ser identificados os itens a serem testados, incluindo uma descrição de cada item e sua versão. A granularidade dos itens de teste depende do projeto a ser testado, e podem ser: programas, módulos, procedimentos, outros (IEEE Standard 829).		<i>Não possui atividade correspondente nesta etapa do Processo.</i>	
		Observação: Na atividade "Especificar Casos de Teste" do processo da Organização, quando os Casos de Testes são definidos é feita menção a diretriz que visa atingir 100% de cobertura para os cenários possíveis de execução. Os itens a serem testados são especificados conjuntamente com a especificação dos Casos de Testes.	
		Etapa do processo em que a Atividade é realizada:	A tarefa de identificação dos itens de teste é realizada na atividade "Planejar Testes" do processo da organização. Porém não é definido um critério de cobertura que considere quais itens e/ou características não serão testadas.
Possibilidade de Melhoria:			
A definição de uma atividade específica para iniciar a caracterização dos testes, identificando os requisitos críticos e o critério de cobertura adequado para as necessidades do projeto pode dotar o processo de teste da característica ágil de convergência, na qual os riscos são atacados pró-ativamente.			
Atividade:		Atividade:	
1.2 Planejar Estratégia para Testar o Produto		<i>Processo da Organização não possui atividade correspondente.</i>	NEQ
Descrição:		Descrição:	
Serão especificadas as decisões do gerente de teste em relação à escolha de técnicas, ferramentas e critério para aceitação ou rejeição de um item de teste em um projeto. Segundo o IEEE STD 829 (1998), neste momento, serão descritas as seguintes informações: Estratégia a ser seguida (níveis de teste: funcionais, integração e sistema) e o Critério de aprovação/rejeição de um item de teste.		<i>Não possui atividade correspondente no Processo.</i>	
		Etapa do processo em que a Atividade é realizada:	As tarefas associadas a esta atividade NÃO são realizadas em nenhuma outra etapa do processo.
Possibilidade de Melhoria:			

A inclusão de uma atividade específica para planejamento da estratégia nesta etapa no Processo pode antecipar o mapeamento das necessidades de teste do projeto a ser desenvolvido. Na estratégia de Teste o gerente responsável pelos testes define os níveis de teste que serão realizados para o projeto (unidade, integração, sistema, aceitação, instalação), escolhe a estratégia de construção e testes (*top-down*, *bottom-up*, *sandwiche*) para organizar como será realizada a integração das várias unidades desenvolvidas e a sequência de testes. Também deverá ser definida a estratégia para re-teste em que o Gerente de Teste deverá definir as técnicas de re-teste: (a) Teste de regressão a ser realizado quando a implementação de mudanças solicitadas podem introduzir novos defeitos no software. (b) *Smoke Testing* utilizado como uma estratégia de teste que suporta integração incremental, quando o software é gerado diariamente contendo os novos componentes integrados e testados. O processo da organização contempla a execução de 3 níveis de teste: unidade, integração e funcional, independente dos requisitos de teste e riscos envolvidos.

Atividade:	Atividade:	
1.3 Definir atividades de Teste	Especificar EAP do Projeto	EQU
Descrição:	Descrição:	
Identificar o conjunto de atividades, suas relações e artefatos necessários para preparar (planejar) e executar os testes.	Na fase de planejamento do Projeto o líder cria a Estrutura Analítica do Projeto em que são explicitadas as atividades de Projeto de Testes (atividade denominada no processo da organização "Planejar Testes") e as atividades de Execução dos Testes. As atividades de Teste são planejadas conjuntamente com as atividades de desenvolvimento.	
Atividade:	Atividade:	
1.4 Planejar Recursos Humanos	Especificar o Plano de RH	EQU
Descrição:	Descrição:	
O gerente de teste baseando-se no conhecimento adquirido sobre a equipe de teste, auxiliado por uma caracterização dos membros que compõem a equipe de teste, deve realizar a seleção da equipe responsável pelos testes no projeto corrente.	O líder de projeto avalia de acordo com os requisitos e riscos do projeto quais as competências a serem utilizadas na construção e nos testes (a relação de competências associadas a cada perfil está descrita em registro específico REG 6.2.1_b, a relação entre perfil e cargo está descrita em Lista de cargos).	
Atividade:	Atividade:	
1.5 Planejar Recursos Físicos	Especificar o Plano de Custos	EQU
Descrição:	Descrição:	
Considerando as necessidades para a criação de um ambiente para projeto e execução dos testes para o projeto, deverá ser planejado dos recursos físicos a serem utilizados durante os testes, como por exemplo, computadores, impressoras, instalação de redes de computadores, ferramentas de apoio aos testes, etc.	Quando houver necessidade de recurso adicional de HW/SW além do definido em documento (lista) contendo os recursos alocados a cada colaborador, este recurso será explicitado no plano de custos do Projeto para que seja incluído no orçamento (de acordo com diretrizes do Processo de Gerencia de Projeto).	
Atividade:	Atividade:	
1.6 Planejar Tempo para Teste	Especificar EAP do Projeto	EQU
Descrição:	Descrição:	

O gerente de teste deverá definir as datas de início e conclusão de cada atividade do processo de teste. Ao final, será obtido um cronograma dos testes no projeto corrente.	Para cada atividade de desenvolvimento do produto explicitada na Estrutura Analítica do Projeto será definida a data de início e conclusão da atividade.	
Atividade:	Atividade:	
1.7 Planejar Custos	Especificar o Plano de Custos	EQU
Descrição:	Descrição:	
O gerente de teste, baseando-se no planejamento realizado até o momento para o projeto (descrito no Plano do Projeto) e nos recursos identificados para os testes, realiza a estimativa dos custos associados aos testes a serem realizados, considerando as características do software a ser testado e as atividades definidas.	Os custos financeiros são controlados pela direção. No projeto, o acompanhamento será baseado na alocação h/h. Quando há necessidade de aquisição de HW/SW o custo é definido no plano de custos do projeto.	
Atividade:	Atividade:	
1.8 Especificar necessidade de Treinamento	Especificar o Plano de Treinamento	EQU
Descrição:	Descrição:	
Durante esta atividade, deverão ser identificados, se pertinente, os treinamentos necessários sobre algum assunto ou tecnologia para a realização dos testes no projeto corrente. A execução desta atividade não é obrigatória. Isso depende exclusivamente do projeto que está sendo realizado.	Na fase de planejamento do Projeto o líder verifica se há necessidade de treinamento, em caso positivo um registro é aberto solicitando treinamento para os participantes.	
Atividade:	Atividade:	
1.9 Identificar Riscos	Especificar o Plano de Riscos	EQU-
Descrição:	Descrição:	
O gerente de teste deve realizar a identificação dos principais riscos associados aos testes de software para o projeto corrente, indicando a probabilidade de ocorrência, o nível de impacto no processo de software, um plano para mitigação dos riscos e um plano de contingência.	Na estratégia de riscos definida no processo de desenvolvimento da organização está registrado um único risco associado ao planejamento e execução dos testes, denominado "testes incorretos", cujas causas apontadas são: <u>a) Especificação de requisitos incompleta.</u> <u>b) Elaboração incorreta dos casos de testes.</u> Na atividade de monitoramento e controle do projeto realizada ao longo da execução das 3 fases que compõem o processo de desenvolvimento, os riscos são monitorados, e um relatório é emitido, registrando as eventuais ocorrências, a análise de causas, e as ações encaminhadas para mitigar ou conter o risco. A qualidade dos Casos de Teste é avaliada a partir da aplicação de um checklist desenvolvido para este fim. Existem riscos comuns às atividades de teste e atividades de desenvolvimento e que são monitorados através da estratégia de risco atual, é o caso do risco equipe indisponível e cliente insatisfeito.	

Possibilidade de Melhoria:

Algumas questões formuladas no checklist dão margem a interpretações subjetivas e por vezes não estão associadas a critérios objetivos. Por exemplo, ao invés de definir um critério de cobertura capaz de ser avaliado objetivamente é incluída no checklist uma questão se os casos de teste possuem boa cobertura em relação aos requisitos.

2. Macro-Atividade Projetar Testes (Processo Padrão)

2		
Macro-Atividade de Teste Processo Padrão	Macro-Atividade de Teste do Processo da Organização	Classificação
Projetar Testes	Planejar Testes	EQU-<t>
Descrição (Projetar Testes - Processo Padrão):	Descrição (Planejar Testes – Processo da Organização):	
Nesta atividade deverão ser detalhadas as técnicas e estratégias de testes que serão utilizadas para cada item de teste. Ao final desta atividade estarão definidos os procedimentos e Casos de Testes para cada item a ser testado e que serão especificados na próxima atividade do Processo de Teste.	O projeto de testes é implementado no processo da organização para cada nível de teste (Unidade, Integração e Sistema) e para cada um destes níveis o nome da atividade inicia com a palavra “Planejar”. A especificação dos casos de Teste é realizada em estratégia bottom-up, iniciando a identificação dos Casos de Testes de Unidade para depois Planejar os Testes de Integração e por ultimo os Testes de Sistema (referenciados no processo da organização como Testes do Produto). Esta estratégia não está em conformidade com a estratégia sugerida pelo padrão IEEE-829: Ao descrever o processo de desenvolvimento de testes é sugerida a adoção de uma estratégia top-down em que são projetados os testes de aceitação, testes de sistema, testes de integração e posteriormente os testes de componentes.	
	Observações complementares:	Nem todas as atividades que constituem esta macro-atividade do Processo padrão estão modeladas no processo da organização, já outras são realizadas em outra etapa do ciclo de vida de Testes.
Artefatos de entrada:	Plano de Testes.	Artefatos de entrada: Especificação Técnica, Documento de Especificação de Requisitos, Plano de Integração do Produto, Documento de Arquitetura.

Artefatos de saída:	Especificação de Projeto de Teste (padrão IEEE-829)	Artefatos de saída:	Plano de Teste de Unidade, Plano de Teste de Integração, Plano de Teste do Produto.
Atividade:	Atividade:		
2.1 Selecionar Item de Teste e Características.	<i>Processo da Organização não possui atividade correspondente nesta etapa do ciclo de vida do Teste.</i>		EQU-t<>
Descrição:	Descrição:		
Durante esta atividade, devem ser selecionados os itens de teste e as características (ou combinações de características) que serão avaliadas no projeto dos testes em questão. Os testes para cada característica ou combinação de características devem ser projetados isoladamente.	<i>Não possui atividade correspondente nesta etapa do Processo.</i>		
	Etapa do processo em que a Atividade é realizada:	Na execução dos Projetos a atividade de seleção dos itens de teste é realizada nas atividades "Planejar Testes de Unidade", "Planejar a Integração do Produto e os Testes de Integração", "Planejar Testes do Produto" do processo. Porém esta atividade não está documentada no processo da Organização.	
Possibilidade de Melhoria:			
A definição de uma atividade para tornar explícito o trabalho de seleção do item de teste e das características a serem testadas documenta a necessidade de sua execução e poderá prover orientação aos membros da equipe de Teste.			
Atividade:	Atividade:		
2.2 Descrever Estratégia para o Projeto de Teste	<i>Processo da Organização não possui atividade correspondente.</i>		NEQ
Descrição:	Descrição:		
Durante esta atividade deve ser especificada a estratégia descrita no plano de teste para avaliar o item de teste e as características selecionadas na atividade anterior, incluindo as técnicas e ferramentas específicas a serem utilizadas. Isso consiste em um detalhamento do que foi descrito superficialmente durante o planejamento dos testes.	<i>Não possui atividade correspondente no Processo.</i>		
	Etapa do processo em que a Atividade é realizada:	As tarefas associadas a esta atividade NÃO são realizadas em nenhuma outra etapa do processo.	
Possibilidade de Melhoria:			

A implementação de uma atividade no processo destinada à definição da técnica e ferramenta a ser utilizada em cada item e característica poderá contribuir para melhorar a qualidade dos testes e a racionalização do uso dos recursos disponíveis.		
Atividade:	Atividade:	
2.3 Identificar Casos e Procedimentos de Teste	Planejar Testes de Unidade, Planejar a Integração do Produto e Testes de Integração, Planejar Testes de Produto.	EQU<t>-
Descrição:	Descrição:	
Durante esta atividade deve ser identificado o conjunto de casos e procedimentos de teste para avaliar o item de teste e a característica selecionada para o projeto de teste em questão.	O analista de Teste antes de iniciar a atividade de especificação dos procedimentos e dos Casos de Teste identifica os conjuntos de Caso de Testes e procedimentos a serem desenvolvidos.	
Possibilidade de Melhoria:		
A definição de uma atividade específica no processo de teste com o objetivo de identificar cada Caso e Procedimento de Teste poderá apoiar as atividades de projeto e construção dos Casos de Testes.		
Atividade:	Atividade:	
2.4 Priorizar Casos e Procedimentos de Teste	Planejar Testes de Unidade, Planejar a Integração do Produto e Testes de Integração, Planejar Testes de Produto.	EQU<t>-
Descrição:	Descrição:	
A ordem de prioridade dos casos e procedimentos de teste deve ser definida baseando-se em alguns fatores, como: os riscos associados aos itens de teste definidos durante a macro-atividade "Planejar Testes" e o cronograma dos testes.	Segundo entrevistas com a equipe SEPG, o analista de Teste durante a execução da atividade de especificação dos procedimentos e Casos de Testes ao identificar cada Caso de Teste deverá avaliar a criticidade dos Casos conjuntamente com o analista de sistemas.	
Possibilidade de Melhoria:		
A definição de uma atividade específica no processo de teste destinada a definir a prioridade de execução dos Casos de Teste poderá contribuir para que as características mais críticas do produto sejam testadas, e assegurar que será definida uma sequência de projeto e execução dos Casos de Testes de acordo as necessidades de geração de massa de dados e criticidade dos elementos a serem testados.		
Atividade:	Atividade:	
2.5 Especificar Critério de Aprovação/Rejeição da Característica	<i>Processo da Organização não possui atividade correspondente.</i>	NEQ
Descrição:	Descrição:	

Durante esta atividade, devem ser especificados os critérios a serem utilizados para determinar, exclusivamente, quando a característica (ou a combinação de características) selecionada para o projeto de teste será aprovada ou rejeitada.	<u>Não possui atividade correspondente no Processo.</u>
Possibilidade de Melhoria:	
A definição de uma atividade específica destinada a definir um critério de aprovação/rejeição apoia a definição dos resultados esperados para cada Caso de Teste a ser realizada na próxima macro-atividade do Processo Padrão.	

3. Macro-Atividade Especificar Casos de Testes (Processo Padrão)

3			
Macro-Atividade de Teste Processo Padrão	Macro-Atividade de Teste Processo de Teste do Processo da Organização		Classificação
Especificar Casos de Teste	Planejar Testes		EQU-/EQU+
Descrição (Especificar Casos de Testes - Processo Padrão):		Descrição (Planejar Testes - Processo da Organização):	
Nesta atividade o Projetista de Teste deverá especificar os casos de testes, descrevendo a condição a ser avaliada, os dados de entrada e os resultados esperados. Também deverão ser definidos os recursos necessários para execução, suas restrições e dependências com outros casos de Teste.		Durante a realização das atividades "Planejar Testes de Unidade", "Preparar Integração e Planejar Testes de Integração" e "Planejar Testes de Produto" o analista de Testes define os dados de entrada, os resultados esperados e recursos necessários para execução. Estas atividades são executadas conjuntamente com as atividades de Identificação dos itens a serem testados, o conjunto de características de cada item, a priorização dos testes.	
Artefatos de entrada:	Especificação de Projeto de Teste (padrão IEEE-829)	Artefatos de entrada:	Especificação Técnica, Documento de Especificação de Requisitos, Plano de Integração do Produto , Documento de Arquitetura
Artefatos de saída:	Especificação de Caso de Teste (padrão IEEE-829)	Artefatos de saída:	Plano de Teste de Unidade, Plano de Teste de Integração, Plano de Teste do Produto
Observações:	Como o trabalho de caracterização e especificação dos Casos de Teste no processo da organização é realizado pela mesma macro-atividade do processo da organização, os artefatos de entrada e saída permanecem os mesmos da etapa anterior.		
Atividade:	Atividade:		

3.1 Caracterizar Caso de Teste	Planejar Testes de Produto	EQU
Descrição:	Descrição:	
Durante esta atividade, devem ser identificadas as informações sobre o caso de teste a ser especificado, como: item de teste ao qual está associado, indicando o requisito e projeto de teste ao qual se refere e característica do produto que pretende avaliar. A partir dessas informações, é possível manter a rastreabilidade entre cada caso de teste e os elementos do projeto correspondente (requisitos do software e projeto de teste).	Na etapa de planejamento de dos testes de software o analista de teste é responsável por desenvolver e documentar, para cada requisito de software, um conjunto de casos de teste (entradas, saídas e critérios de teste) e procedimentos de teste. Os Casos de Testes são cadastrados na ferramenta <i>TestLink</i> associados a requisitos e Casos de Uso.	
Atividade:	Atividade:	
3.2 Especificar Entrada e Resultado Esperado	Planejar Testes de Produto	EQU-
Descrição:	Descrição:	
Casos de teste são compostos por valores de entrada e comportamentos e resultados esperados. Esta atividade define esses valores.	As entradas e saídas são especificadas nesta etapa do processo.	
Possibilidade de Melhoria:		
Incluir na documentação das atividades de planejamento dos testes (integração) uma referencia explicita à especificação de entrada e resultado esperado.		
Atividade:	Atividade:	
3.3 Definir Recursos para o Caso de Teste	Planejar Testes de Produto	EQU-
Descrição:	Descrição:	
Durante esta atividade, serão definidos os recursos necessários para a execução do caso de teste específico, como, por exemplo, hardware ou software. Os recursos devem ter sido especificados durante a macro-atividade "Planejar Testes".	Na especificação dos Casos de Testes quando necessário são informados as tabelas/dados e recursos de hw necessários, porém não existe uma atividade destinada para a realização desta tarefa.	
Possibilidade de Melhoria:		
Na atividade de planejamento dos testes deverão ser consideradas as necessidades de recursos de software como tabelas e outros elementos do <i>testware</i> facilitadores na execução dos testes. (como por exemplo, a geração de bancos de dados semente e outros recursos como <i>stubs, drivers e mocks</i>)		
Atividade:	Atividade:	
3.4 Especificar Restrições de Uso	<i>Processo da Organização não possui atividade correspondente.</i>	NEQ
Descrição:	Descrição:	
Durante esta atividade serão descritas possíveis restrições no caso de teste	<u><i>Não possui atividade correspondente no Processo.</i></u>	

em questão. Essas restrições podem envolver configuração, procedimentos para obtenção dos resultados e conclusão do caso de teste.		
Possibilidade de Melhoria:		
A definição de uma atividade específica no processo de teste destinada a especificar as restrições de execução dos CTs apoia o desenvolvimento e planejamento dos Procedimentos de Teste, atividade modelada na próxima macro-atividade do Processo Padrão.		
Atividade:	Atividade:	
3.5 Definir Dependências entre Casos de Teste	Planejar Testes do Produto	EQU-
Descrição:	Descrição:	
Identificar quais casos de teste devem ser executados antes do caso de teste em questão. Deve ser especificada a origem desta dependência.	Na atividade "Planejar Testes do Produto" as dependências são identificadas na fase de planejamento dos Testes, porém não existe referencia formal a esta atividade no processo de teste da organização.	
Possibilidade de Melhoria:		
A definição de uma atividade específica no processo de teste destinada a registrar eventuais dependências entre os CTs apoia o desenvolvimento e planejamento dos Procedimentos de Teste, atividade modelada na próxima macro-atividade do Processo Padrão.		
Atividade:	Atividade:	
<i>Processo Padrão não possui atividade correspondente.</i>	Avaliar Conformidade dos Planos de Integração e Testes do Produto	EQU+
Descrição:	Descrição:	
<i>Não possui atividade correspondente no Processo</i>	Avaliar o Plano de Testes do Produto composto pelos Planos de testes de Unidades, Integração e Software e o Plano de Integração do Produto quanto ao seu formato, segundo critérios de qualidade pré-estabelecidos pelo GQPP.	
Possibilidade de Melhoria:		
N/A		
Atividade:	Atividade:	
<i>Processo Padrão não possui atividade correspondente.</i>	Avaliar o Plano de Testes do Produto	EQU+
Descrição:	Descrição:	
<i>Não possui atividade correspondente no Processo</i>	Avaliar o Plano de Testes do Produto, contemplando os testes de unidade, integração e do software, usando um checklist de questões definidas pelo Processo de Verificação.	

Possibilidade de Melhoria:
N/A

4. Macro-Atividade Definir Procedimentos de Testes (Processo Padrão)

4		
Macro-Atividade de Teste Processo Padrão	Macro-Atividade de Teste do Processo da Organização	Classificação
Definir Procedimentos de Teste	Planejar Testes de Software	EQU-
Descrição (Definir Procedimentos de Testes - Processo Padrão):		Descrição (Planejar Testes de Software- Processo da Organização):
Nesta atividade o projetista de teste descreve os passos necessários para a execução de um ou um grupo de casos de teste. Um procedimento de teste precisa ser caracterizado com informações sobre o seu objetivo e requisitos para a sua execução, além dos passos a serem seguidos durante os testes.		Durante a realização da atividade "Planejar Testes de Software" o analista de Testes identifica os passos necessários para execução dos Casos de Testes e os agrupa em suítes de Teste, sendo cada Suíte de Teste associada a um conjunto de objetivos.
Artefatos de entrada:	Especificação de Caso de Teste (padrão IEEE-829)	Artefatos de entrada: Especificação Técnica, Documento de Especificação de Requisitos, Plano de Integração do Produto, Documento de Arquitetura.
Artefatos de saída:	Especificação de Procedimentos de Teste (padrão IEEE-829)	Artefatos de saída: Plano de Teste de Unidade, Plano de Teste de Integração, Plano de Teste do Produto.
Observações:	Como o trabalho de definição dos procedimentos de teste é realizado pela mesma macro-atividade do processo da organização, os artefatos de entrada e saída permanecem os mesmos da etapa anterior.	
Atividade:	Atividade:	
4.1 Definir Objetivos	Planejar Testes de Produto	EQU-
Descrição:		Descrição:
Durante esta atividade, devem ser identificados os objetivos para um procedimento de teste, indicando quais itens de teste estão associados ao procedimento em questão.		Apesar dos objetivos estarem cadastrados na ferramenta DTS (<i>Defect Track System</i>) não existe referência formal a esta atividade no processo da organização.
Possibilidade de Melhoria:		

A definição de uma atividade destinada a definir os objetivos de cada procedimento poderá apoiar a avaliação dos resultados da execução destes procedimentos.		
Atividade:	Atividade:	
4.2 Definir Requisitos para a Execução do Procedimento	<i>Processo da Organização não possui atividade correspondente.</i>	NEQ
Descrição:	Descrição:	
Durante esta atividade, serão identificados quais os requisitos para a execução deste procedimento de teste. Isto pode incluir procedimentos que são pré-requisitos, necessidades do ambiente de teste ou habilidades especiais do testador para a execução do procedimento.	<i>Não possui atividade correspondente no Processo.</i>	
Possibilidade de Melhoria:		
A definição de uma atividade destinada à definição dos requisitos para execução do procedimento apoia a macro-atividade de execução dos testes.		
Atividade:	Atividade:	
4.3 Descrever Passos	<i>Processo da Organização não possui atividade correspondente.</i>	NEQ
Descrição:	Descrição:	
Durante esta atividade, devem ser descritos todos os passos a serem seguidos para a execução do procedimento de teste: a) Configurar: ações necessárias para preparar a execução do procedimento; b) Iniciar: ações necessárias para iniciar a execução do procedimento; c) Executar: ações necessárias durante a execução do procedimento; d) Medir: descreve como as medições dos testes serão realizadas; e) Suspender: ações necessárias para suspender os testes; f) Reiniciar: ações necessárias para reiniciar os testes que foram suspensos; g) Parar: ações necessárias para finalizar os testes; h) Contingências: ações necessárias para lidar com eventos anormais.	<i>Não possui atividade correspondente no Processo.</i>	
Possibilidade de Melhoria:		
A definição de uma atividade específica destinada a descrever os passos necessários para a execução dos procedimentos apoia a macro-atividade de execução dos testes.		
Atividade:	Atividade:	
4.4 Construir Procedimento de Teste (opcional)	<i>Processo da Organização não possui atividade correspondente.</i>	NEQ
Descrição:	Descrição:	

Durante esta atividade os procedimentos de testes devem ser construídos (por meio da criação de scripts), caso necessite, para serem utilizados durante o subprocesso de execução dos testes.	<u>Não possui atividade correspondente no Processo.</u>
Possibilidade de Melhoria:	
A definição de uma atividade específica no processo de teste destinada a construir os scripts para automatizar os procedimentos apoia a macro-atividade de projeto dos testes.	

5. Macro-Atividade Executar Testes (Processo Padrão)

5			
Macro-Atividade de Teste Processo Padrão	Macro-Atividade de Teste do Processo da Organização	Classificação	
Executar Testes	Testes do Produto	EQU+	
Descrição (Executar Testes- Processo Padrão)		Descrição (Testes do Produto - Processo da Organização):	
Nesta macro-atividade o ambiente de teste deverá configurado, os testes serão executados, e cada incidente ocorrido durante a execução deverá ser registrado. Os procedimentos devem seguir o planejado e a execução registrada de forma ser possível repetir os mesmos procedimentos. As atividades a serem realizadas durante a execução dos testes devem ser conduzidas pelos Testadores e são baseadas nas informações contidas nos documentos SWEBOK (ABRAN et al., 2001), Histórico dos Testes (IEEE STD 829, 1998) e Relatório de Incidente de Teste (IEEE STD 829, 1998).		Realizar os testes do produto com base nos Casos de Teste contidos no Plano de Testes. Deve ser garantido que a implementação de cada requisito do produto seja testada para conformidade. Os resultados dos testes devem ser documentados no Relatório de Testes do Produto. Para as falhas encontradas deverá ser criado um Plano de Ação (ticket).	
Artefatos de entrada:	Especificação de Procedimentos de Teste (padrão IEEE-829)	Artefatos de entrada:	Artefato de Software Executável, Plano de Testes do Produto.
Artefatos de saída:	Histórico dos Testes (padrão IEEE-829), Relatório de Incidente dos Testes (padrão IEEE-829)	Artefatos de saída:	Relatório de Testes do Produto, Plano de Ação.
Atividade:		Atividade:	
5.1 Configurar Ambiente de Teste	Estabelecer o Ambiente de Integração do Produto	QUE	
Descrição:		Descrição:	
O ambiente de teste deve ser compatível com ambiente de desenvolvimento		Nesta atividade o Analista de Produção de posse do Plano de Integração do	

de software. Ele deve facilitar a execução e controle dos casos e procedimentos de teste, bem como registrar os resultados obtidos. Durante esta atividade, o ambiente de teste deve ser ajustado para a realização dos testes previamente planejados.	produto avalia os componentes de software e hardware necessários para a realização dos testes de integração e prepara o ambiente de acordo com o especificado.	
Possibilidade de Melhoria:		
N/A		
Atividade:	Atividade:	
5.2 Executar Procedimentos de Teste	Codificar e Testar os Componentes do Produto	EQU-
Descrição:	Descrição:	
Durante esta atividade devem ser executados os procedimentos de teste previamente definidos, e para cada procedimento de teste a ser realizado será realizada uma comparação entre os resultados obtidos por meio da execução e o resultado esperado, definido durante o processo de planejamento. Cada evento realizado deve ser registrado no documento Histórico dos Testes.	São três atividades no processo da Organização responsáveis pela execução dos testes. A primeira execução ocorre conjuntamente com a construção dos artefatos de software e é responsável pela realização dos testes de unidade e integração. Esta atividade é realizada pelo desenvolvedor.	
	Artefatos de entrada:	Documento de Especificação de Requisitos, Especificação Técnica, Plano de Testes de Unidade.
	Artefatos de saída:	Código de unidades, Relatório de Testes de Unidade, Plano de Ação.
	Atividade:	
	Integrar o Produto e Testar a Integração	EQU
	Descrição:	
A segunda execução de teste ocorre na fase de integração das unidades e componentes de software quando as agregações são testadas à medida que são integradas. Nesta etapa são realizadas as integrações de itens de software a hardware e a outros sistemas se necessário. Os resultados da integração e dos testes são documentados. A meta é assegurar que o item integrado está pronto para os testes do produto, cujo objetivo é confirmar que o produto integrado atinge os requisitos definidos. Caso algum defeito seja identificado, um Plano de Ação deve ser aberto e gerenciado até sua conclusão, de forma a permitir a execução da próxima atividade.		
Artefatos de entrada:	Código de Unidades, Plano de Testes de Integração.	

	Artefatos de saída:	Produto integrado, Relatório de Testes de Integração, Plano de Ação.
	Atividade:	
	Realizar Testes do Produto	EQU
	Descrição:	
	A terceira execução ocorre quando o artefato foi construído, os testes de unidade e integração foram executados. Nesta etapa é realizado o teste de software.	
	Artefatos de entrada:	Artefato de Software Executável, Plano de Testes do Produto.
Artefatos de saída:	Relatório de Testes do Produto, Plano de Ação.	
Possibilidade de Melhoria:		
Avaliar a possibilidade de incluir nos relatórios gerados pelas atividades de execução dos três níveis de teste da organização informações presentes no documento Histórico de Testes propostos pelo padrão IEE829.		
Atividade:	Atividade:	
5.3 Registrar Incidentes de Teste	Registrar homologação interna do produto	EQU-
Descrição:	Descrição:	
Durante esta atividade devem ser registrados os incidentes ocorridos durante a execução dos procedimentos de teste, indicando diversas informações sobre o incidente, como: resultados obtidos e esperados, a anomalia ocasionada no sistema, o passo do procedimento de teste em que ocorreu o incidente e as pessoas envolvidas nesse procedimento.	Registrar a homologação interna do produto e Avaliar os resultados da inspeção e dos testes de unidade, integração e do produto e, se for o caso, definir Planos de Ação para correção dos eventuais problemas encontrados. Nesta atividade é aplicado um checklist para avaliar o resultado dos testes realizados. As questões visam a verificação se os resultados registrados estão de acordo com os objetivos definidos no plano de testes em termos de acurácia e consistência.	
Possibilidade de Melhoria:		
Avaliar a possibilidade de incluir nos relatórios gerados pelas atividades de execução dos três níveis de teste da organização informações presentes nos relatórios propostos pelo padrão IEE829.		

6. Macro-Atividade Analisar Resultados dos Testes (Processo Padrão)

6		
Macro-Atividade de Teste Processo Padrão	Macro-Atividade de Teste do Processo da Organização	Classificação
Analisar Resultados dos testes	Encerramento do Desenvolvimento	EQU-
Descrição (Analisar Resultados dos Testes- Processo Padrão)	Descrição (Encerramento do Desenvolvimento – Processo da Organização)	
O Gerente de teste deverá avaliar os resultados dos testes para determinar se obtiveram sucesso, além de coletar medidas específicas. A análise dos resultados dos testes permite a uma organização prover um entendimento sobre o seu processo de desenvolvimento identificando pontos de falhas durante o desenvolvimento, e conseqüentemente prover melhorias no seu processo. As atividades a serem realizadas durante a definição dos procedimentos de teste devem ser conduzidas pelo Gerente de Teste e são baseadas nas informações que compõem o documento Relatório de Resumo de Teste, descrito no IEEE STD 829 (1998).	O objetivo do Encerramento do Desenvolvimento é avaliar o projeto após sua conclusão, registrando seus resultados, lições aprendidas, melhores práticas, avaliação da equipe, avaliação dos processos, com indicação de oportunidades de melhoria; avaliação dos ativos.	
Artefatos de entrada:	Relatório de Incidentes de Teste (padrão IEEE-829), Histórico dos Testes (padrão IEEE-829)	Artefatos de entrada: Questionário de Avaliação da Satisfação do Cliente
Artefatos de saída:	Relatório de resumo dos Testes(padrão IEEE-829)	Artefatos de saída: Questionário de Avaliação da Satisfação do Cliente, Dados para atualização do Repositório de Estimativas
Atividade:	Atividade:	
6.1 Resumir Testes	Registrar homologação interna do produto	EQU-
Descrição:	Descrição:	
Durante esta atividade deve ser realizado um resumo da avaliação de cada item de teste definido, indicando uma descrição do ambiente em que foram realizados os testes e os incidentes ocorridos.	Registrar a homologação interna do produto e Avaliar os resultados da inspeção e dos testes de unidade, integração e do produto e, se for o caso, definir Planos de Ação para correção dos eventuais problemas encontrados. Nesta atividade é aplicado um checklist para avaliar o resultado dos testes realizados. As questões visam uma verificação se os resultados registrados estão de acordo com os objetivos definidos no plano de testes em termos de acurácia e consistência	

Possibilidade de Melhoria:	
Não é realizada uma avaliação dos incidentes ocorridos durante os testes. Esta avaliação poderia ser realizada na etapa de registro da homologação interna do produto para que pudesse ser analisada a causa raiz destes incidentes para melhoria do processo e compartilhamento de conhecimento entre os envolvidos.	
Atividade:	Atividade:
6.2 Registrar Dados dos Testes	Registrar Métricas EQU+
Descrição:	Descrição:
Durante esta atividade devem ser armazenados os dados obtidos com a realização dos testes, incluindo as informações sobre o seu planejamento (itens de teste, recursos utilizados, cronograma, casos e procedimentos de teste), os incidentes detectados e seu impacto, a fim de prover informações sobre testes para viabilizar melhorias nos testes em projetos futuros, além de disponibilizar produtos gerados ao longo do processo de testes para serem reutilizados em projetos futuros.	São três atividades no processo da Organização responsáveis pelo registro dos resultados e avaliação dos processos de software. A primeira atividade é a coleta das métricas do projeto, executada em cada fase do processo padrão da organização. Esta atividade é realizada pelo líder do projeto e participam o Grupo de medição, o Grupo de Garantia da Qualidade do Processo e do Produto (GQPP), Grupo de Gerência de Configuração (GCO).
Artefatos de entrada:	Registro das atividades de projeto realizadas.
Artefatos de saída:	Metas registradas
Atividade:	Atividade:
Coletar Dados para Avaliação do Projeto	EQU+
Descrição:	Descrição:
	A segunda atividade é responsável pela coleta dos dados de medição do projeto (esforço realizado, esforço estimado, etc.), dados de satisfação do usuário/cliente (questionários de satisfação), confirmar o tamanho do projeto e fazer a preparação para a avaliação post-mortem e para a avaliação da equipe.
Artefatos de entrada:	Questionário de Avaliação da Satisfação do Cliente, Registro das atividades de projeto.
Artefatos de saída:	Questionário de Avaliação da Satisfação do Cliente preenchido.
Atividade:	Atividade:
Realizar Avaliação Post-Mortem	EQU-
Descrição:	Descrição:

	A terceira atividade tem como objetivo realizar uma avaliação retrospectiva ao final do projeto e documentar seus resultados, contemplando a avaliação dos processos, com indicação de oportunidades de melhoria, indicação de lições aprendidas e avaliação dos ativos. Realizar a reunião de encerramento do projeto.	
	Artefatos de entrada:	Questionário de Avaliação da Satisfação do Cliente
	Artefatos de saída:	Dados para atualização do Repositório de Estimativas e Relatório de Avaliação Post-Mortem contendo para cada processo, pontos fortes, pontos fracos e considerações gerais. Se houver melhoria, abrir ticket; indicação do resultado da implantação de tickets ao longo do período do projeto, com seus efeitos (positivos ou negativos); e identificação de oportunidades de melhoria (com ticket associado).
Possibilidade de Melhoria:		
Avaliar a possibilidade de incluir nos relatórios gerados pelas atividades de execução (dos três níveis de teste da organização) informações presentes no documento Histórico de Testes propostos pelo padrão IEE829.		

E.3 Artefatos dos Processos.

Conteúdo Artefato Plano de Teste Processo Padrão	Identificador Único:	Conteúdo Artefato Plano de Testes Processo Organização:	
	Status:		
	Data Criação:		Data Criação:
	Data Conclusão:		
	Escopo/Requisitos		Escopo/Requisitos:
	Referencias a UCs/Plano Projeto:		Referencias a Requisitos Funcionais:
	Itens de Software a serem testados:		Itens de Software a serem testados:
Critério de Aprovação:			

	Características a serem testadas:		
	Níveis de Teste a serem adotados:		Todos os Níveis de Teste são utilizados em cada projeto:
	Necessidades de SW e HW:		Necessidades de SW e HW.
	Recursos Humanos p/ Teste:		Recursos Humanos p/ Teste.
	EAP atividades de Teste Processo Padrão.		EAP atividades de Teste: Projetar Testes Unidade, Projetar Testes Integração, Projetar Testes Software, Executar Testes.
Riscos e Contingência			Riscos e Contingencia.

Conteúdo do Artefato "Especificação de Projeto de Testes"	Identificador Único:	Não existe artefato a ser produzido para a etapa de Especificação de Projeto de Teste no Processo da Organização utilizada no Estudo de Caso.
	Status:	
	Dt Criação:	
	Dt Conclusão:	
	Itens do Software a serem testados:	
	Características a serem testadas:	
	Abordagem/Critério de Cobertura: (exemplo: Todos os Fluxos Principais e Alternativos dos UCs de cada item a ser testado)	
	Relação de Casos de Testes:	
	Relação de Procedimentos:	
	Níveis de Teste a serem adotados:	
Critério de Aprovação/Rejeição do Teste: Os testes das funções definidas serão considerados aprovados (PASSED) somente se nenhum incidente for registrado durante a execução dos testes.		

Conteúdo do Artefato "Especificação de Caso de Testes"	Identificador Único:	Conteúdo do Artefato "Especificação de Caso de Testes" Organização Estudo de Caso.	Identificador Único:
	Status:		Status:
	Data Criação:		Data Criação:
	Data Conclusão:		Data Conclusão:
	Item de Software a ser testado:		Item de Software a ser testado:
	Entradas do Caso de Teste:		Entradas do Caso de Teste:
	Resultados Esperados:		Resultados Esperados:
	Necessidade de ambiente do Caso de Teste:		Necessidade de ambiente do Caso de Teste:
	Dependência com outros Casos de Teste:		Dependência com outros Casos de Teste:
	Dependência com outros Casos de Teste:		Dependência com outros Casos de Teste:
Conteúdo do Artefato "Especificação de Procedimento de Testes"	Identificador Único:	Conteúdo do Artefato "Especificação de Caso de Testes" Organização Estudo de Caso.	Identificador Único:
	Status:		Status:
	Data Criação:		Data Criação:
	Data Conclusão:		Data Conclusão:
	Propósito do Procedimento: (Informa a funcionalidade que o procedimento irá exercitar)		Propósito do Procedimento: (Informa a funcionalidade que o procedimento irá exercitar)
	Passos do Procedimento:		Passos do Procedimento:
Conteúdo do Artefato "Log de Testes"	Identificador Único:	Não existe artefato a ser produzido para Log de Teste no Processo da Organização utilizada no Estudo de Caso. Organização utiliza ferramenta de <i>Bug-Tracker</i> (TRAC) para registro dos incidentes de Teste. Testador registra na ferramenta <i>Testlink</i> resultado de Teste e quando este apresenta uma inconsistência cadastra um Ticket no TRAC sendo gerado um plano de ação.	
	Status:		
	Data de Início:		
	Data de Término:		
	Entradas de Atividades e Eventos:		
	Passos do Procedimento:		
Conteúdo do Artefato "Incidentes de Testes por <i>Trial</i> "	Identificador Único:	Não existe artefato a ser produzido para Incidente de Teste no Processo da Organização utilizada no Estudo de Caso. Organização utiliza ferramenta	
	Nível de Severidade:		

	Data da ocorrência:	Testlink onde estão registradas as ocorrências de execução dos Casos de Testes. Porém não utiliza uma classificação de severidade das falhas encontradas.
	Descrição:	
	Entradas:	
	Resultados Esperados:	
	Resultados Obtidos:	
	Anomalias:	
	Tentativas de repetição:	

Conteúdo do Artefato "Relatório de Resumo dos Testes"	Identificador Único:	Conteúdo do Artefato "Relatório de Testes" Organização Estudo de Caso	Identificador Único:
	Itens de Teste:		Itens de Teste:
	Características:		
	Artefatos produzidos:		
	Relatórios de Incidentes de Teste: (relatórios dos incidentes por <i>trial</i>)		Relatórios de Incidentes de Teste: (relatórios dos incidentes por <i>trial</i>)
	Desvios em relação ao planejamento (atrasos, mudanças necessárias, replanejamentos):		
	Abrangência dos Testes (o que não pode ser testado e a razão para o problema)		
	Análise dos Critérios de aceitação dos itens de Teste (caso exista algum critério não atendido esta informação será registrada)		
	Resumo de EAP das atividades de Teste.		
	Resumo dos resultados: (informa se os critérios de aprovação foram cumpridos)		Resumo dos resultados: (informa se os critérios de aprovação foram cumpridos)

APÊNDICE F – Atas reuniões

Neste apêndice são apresentadas as atas de reunião realizadas na organização do estudo de Caso.

F.1 Apresentação propostas de mudanças para o SEPG

Reunião Grupo de Processo de Engenharia de Software (SEPG)

Data: 19/10/2012

Participantes:

- <Participante 1>(Verificação, Validação e Garantia da Qualidade)
- <Participante 2> (Escritório de Projetos)
- <Participante 3> (Diretor Administrativo-Financeiro e Responsável pela Gerencia de RH)
- <Participante 4> (Responsável pelo SEPG)
- <Participante 5> (Escritório de Projetos)

Ciro Grippi Barbosa Lima (Pesquisador COPPE/UFRJ)

Assuntos:

1- Acompanhamento dos Processos:

1.1- A última reunião do SEPG foi realizada em 21/9/2012.

1.2- Após definição da ordem dos assuntos a serem tratados, Ciro apresentou modelo do processo atual da Informal Informática em versão HTML, contendo todas as atividades, perfis, artefatos e critérios de entrada e saída relativas às 3 fases do processo padrão de desenvolvimento.

Esta versão HTML foi gerada a partir do uso do software *Eclipse FrameWork Composer*-EPF 1.5.1.4, quando toda a documentação contida em arquivo Word da atual versão da Instrução de Procedimento 7.3 foi cadastrada em um repositório padrão SPEM (modelo para desenvolvimento de software da OMG).

Conjuntamente com a versão atual do processo foi mostrada uma proposta de evolução, também em formato HTML, desenvolvida a partir de um modelo, contendo propostas de inserção de novas atividades prescritas em um processo de Teste baseado na especificação IEE-829, e de algumas práticas que visam dotar o processo atual de algumas características ágeis como feedback, facilidade de adaptação à mudanças e diminuição de perdas.

1-3- Após explicação detalhada de cada ponto, houve consenso que as sugestões de inclusão de novas atividades e antecipação de algumas atividades de Teste para fases iniciais do processo são pertinentes. Inclusive, refletindo melhor a realidade dos projetos. Foi então combinado que um estudo será conduzido – com acompanhamento do Grupo de Processos – em que estas mudanças serão experimentadas e avaliadas.

1.4- Quanto à sugestão de junção das atividades “Avaliar Conformidade Plano de Integração e Plano Testes do Produto” e “Avaliar o Plano de Testes do Produto” em uma única atividade o gerente do SEPG comentou que já está em andamento uma proposta de especialização de Processos na qual algumas atividades serão facultativas. Sendo provável que algumas atividades de GQA venham a s ser agrupadas com o intuito de trazer mais leveza à execução

do processo sem perda de qualidade final. Desta forma, o estudo proposto não contemplará esta proposta de mudança.

1.4- Assim, a partir da navegação no EPF e explicação da notação utilizada, 3 alterações principais foram selecionadas:

- Evolução da IT: criar atividade na Fase 1 para Definir Estratégia do Projeto de Teste; e antecipar a Elaboração do Plano de teste para o início da Fase 2
- Revisão do padrão de elaboração dos Casos de Testes
- Adoção do EPF para definição e manutenção da IT 7.3 (em data a ser combinada)

1.5- Com o objetivo de facilitar a adoção do novo padrão de testes será utilizado um Editor para definição dos CTs, com a proposta de padronização de termos.

1.6- Para que sejam observados os eventuais benefícios da adoção destas mudanças será escolhido um Sprint de um dos projetos em curso para avaliação destas propostas,

1.6- Para avançar nas questões, um ticket será aberto por melhoria, visando análise e posterior liberação na próxima baseline (3.12), prevista para 23/11 (avaliar antecipação).

1.7- Duas ações estratégicas associadas a Dimensão Competitiva Excelência de Serviços estão em andamento:

- DES06 - Promover evolução contínua do Processo de Software (melhoria horizontal); e
- DES07 - Definir processos especializados.

1.8- A melhoria contínua está em curso normal, com liberação de baselines a cada 2m. A última baseline foi a 3.11, implantada em 28/9/2012, tendo no escopo 9 tickets.

1.9- Em agosto foi iniciada a definição do processo especializado para Fábrica de Teste e Projetos de Manutenção. Entretanto, devido principalmente ao aumento do número de projetos no período, a definição não foi concluída, o que deve ser retomado.

1.10- Conforme mapeado na reunião de 22/8, junto com a especialização serão definidos critérios/diretrizes de adaptação, com respectivas metas e questões correspondentes. Objetivo: contribuir com a base de medição e com o repositório de estimativa.

2- Relatórios de Situação dos Processos (REG 7.3.a) e Medição (REG 8.4.b)

2.1- Os relatórios de situação e medição 001/2012 estão concluídos, aprovados e tiveram seu resultados apresentados em palestra (31/7).

2.2- Já os relatórios de situação 002, 003 e 004/2012 também foram concluídos, mas terão seus resultados apresentados na próxima palestra mensal.

2.3- Houve atraso no fechamento dos mesmos devido a pendência de finalização dos projetos no <Ferramenta Gp>. Para tal, foram abertas não conformidades específicas, as quais devem ser tratadas pelos responsáveis.

2.4- Já o relatório 005/2012 está em fase final de elaboração, conforme previsto no cronograma de atividades organizacionais.

2.5- A reunião de apresentação dos mesmos à Diretoria será realizada ainda hoje, logo após a reunião do SEPG.

3- Cronograma de Atividades Organizacionais 2012 (REG 7.3.d):

3.1- Cronograma em andamento, com apoio do PMO para monitoração.

3.2- Última versão publicada na baseline 3.10. Atualizar para a próxima baseline (3.12).

3.3- Auditoria do GQPP: Estava planejada para outubro, mas foi remarçada para Dezembro. Confirmar data e escopo a ser avaliado (projetos e relatórios envolvidos).

4- Assuntos Gerais:

4.1- Este mês vai ocorrer o WAMPS 2012 – Workshop Anual do MPS.BR, em Campinas, de 22 a 24/10/2012, com participação de <gerente SEPG>.

4.2- O artigo “Implementação do MPS.BR: Um Relato da Trajetória de Melhoria até o Nível C de Maturidade” foi aceito e será apresentado no evento.

F.2 Reunião para análise dos resultados obtidos

F.2.1 Reunião para avaliação de mudanças a serem incluídas na *baseline* do processo da organização.

----- Forwarded Message -----

From: <gerente SEPG >

To: <Escritorio de Projetos >

Cc: "Ciro Grippi Barbosa Lima" <cblima@cos.ufrj.br>

Sent: Sat, 18 May 2013 10:36:57 -0300

Subject: RES: Modificações sugeridas para o processo padrao.

Srs., para acompanhamento, finalizamos a avaliação do Projeto Piloto para evolução das atividades de Teste.

Anexei a avaliação do SEPG e do PMO, ambas debatidas em 15/5 com o pesquisador e com a equipe envolvida no projeto.

Com isso, os tickets referentes as melhorias propostas serão analisados na próxima reunião do SEPG, visando liberação na baseline 3.15, prevista para 31/maio.

São eles:

#653: Melhoria - VER - Definir Estratégia do Projeto de Teste

#654: Melhoria - VER - Revisar padrão de elaboração dos Casos de Testes

#655: Melhoria - DFP - Adotar EPF para definição e manutenção da IT 7.3

#656: Melhoria - DFP - Adotar Editor para definição dos Casos de Teste.

<gerente SEPG>

-----Mensagem original-----

De: <gerente SEPG>

Enviada em: terça-feira, 1 de janeiro de 2013 16:15

Para: 'Ciro Grippi Barbosa Lima'

Cc: <sepg>

Assunto: RES: Modificações sugeridas para o processo padrao.

Ciro, registrei a análise e implementação dos tickets #653, #654 e #656 no SGQ Informal, considerando que os três serão experimentados no projeto 695 - PB v 2012.4 - DadosPT.

Objetivo: evoluir práticas do Processo de Teste da Informal.

O processo do projeto foi adaptado, com incorporação das atividades de melhoria correspondentes. Agora o mesmo está em avaliação pelo GQPP. A seguir será avaliado pelo PMO e posteriormente pelo cliente.

Estão previstos 4 ciclos/etapas, a saber:

- (1), em novembro/2012;

- (2), em dezembro/2012 e janeiro/2013;

- (3), em fevereiro/2013; e

- (4), em março/2013.

No encontro de amanhã, 2/1, já realizaremos o treinamento nas novas práticas e no editor.

Att, <Gerente SEPG>

-----Mensagem original-----

De: **Ciro Grippi Barbosa Lima** [mailto:cblima@cos.ufrj.br]
Enviada em: segunda-feira, 26 de novembro de 2012 00:39
Para: <gerente SEPG>
Cc: sepg
Assunto: Modificações sugeridas para o processo padrao.

<< Gerente do SEPG>>,

de acordo com o combinado em reunião com o grupo SEPG em 09/11 seguem as alterações a serem realizadas no projeto (todas elas assinaladas no site <http://<Intranet organizacao>:33380/it73pro/>):

1) Na etapa 1 do processo padrão relativa a "Especificação e Planejamento de Projeto" deverá ser incluída uma macro-atividade para caracterização dos itens sob teste e sobre as características (funcionais, desempenho, usabilidade , carga etc) escolhidas e que farão parte do escopo do planejamento de testes.

Atividades que compõem a macro-atividade a ser incluída: Caracterizar testes , Planejar Estratégia para Testar o Produto, Revisar Planejamento Atividades de Teste.

Artefato atualizado: Plano de Testes

--> Ticket #653

2) Na etapa 2 de "Análise Projeto e Construção do Software" na macro-atividade "Planejar Testes" foi incluída a atividade "Projetar Testes" contendo as seguintes sub-atividades:

- "Selecionar Item de Teste e Característica"
- "Descrver Estratégia para o Projeto de Teste"
- "Identificar Casos de Teste e Procedimentos"
- "Priorizar Casos e Procedimentos e Casos de Teste"
- "Especificar Critério de Aprovação e Rejeição da Característica"

Como a atividade de "Planejar Testes" passou a ocorrer em paralelo com as atividades "Preparar Integração do Produto" e "Realizar Especificação Técnica" as atividades de "Avaliar Conformidade do Plano de Integracao do Produto" e "Avaliar o Plano de Testes do Produto" deverão ser executadas ao final da Macro-atividade "Realizar a especificacao tecnica" .

----> Esta não será feita, pois não foi possível contactar o stakeholder para alinhamento.

Artefato atualizado : Plano de testes (com inclusão de novo padrão de Casos de Testes aonde os passos dos procedimentos, resultados esperados e pré-condições são definidos separadamente para preservar a distinção entre dados e passos.

----> Tickets #654 e #656

Qualquer dúvida, podemos marcar reunião para realização do planejamento do projeto que estará sendo utilizado como piloto das modificações a serem avaliadas.

att

APÊNDICE G – Identificação de Cenários e CTs

Neste apêndice são apresentados os Casos de Testes e Cenários identificados na atividade de caracterização de Testes da Fase 1 do Processo Especializado SCRUM, extraídos de tabela do Plano de Testes que faz parte da relação dos artefatos definidos pelo processo de Definição dos Processo Organizacionais

G1 – Tabela de Casos de Uso, Cenários e Casos de Teste

Casos de Uso	Prioridade	Cenário	Caso de Teste
UC. 136 - Visualizar menu principal	3	CEN.001 - Usuário Petrobrás autenticado no PB acessa Base tipo BD EAB/EATG	Conferir se acesso é feito com sucesso
		CEN.002 - Usuário Petrobrás autenticado com qualquer perfil no PB acessa Base tipo SQL COMPLETO	Ausência do link no menu principal “Petrobras”
		CEN.003 - Usuário autenticado com perfil "administrador" PB acessa Base tipo SQL COMPLETO	Exibição do Link “Visões personalizadas”;
UC. 142 - Incluir Proposta de Trabalho	3	Usuário Petrobrás autenticado com permissão de “administrador” pelo Project Builder	Conferir se Guias Gestão e Planejamento ficam disponíveis para o usuário
	3	Usuário Petrobrás autenticado com permissão diferente de “administrador” pelo Project Builder, mas com permissão “Incluir Projetos”	Conferir se a Guia Gestão fica disponível para o usuário; e se a Guia Planejamento está indisponível.
	3	Usuário Petrobrás autenticado com permissão diferente de “administrador” pelo Project Builder, mas sem a permissão “Incluir Projetos”	Conferir se a opção Nova PT do menu principal não é exibida
UC. 143 - Listar Propostas de Trabalho	2	CEN.001 - Usuário autenticado em uma Base EAB EATG solicita a opção "Lista de PT"	Exibição da página Lista de PT no padrão PB
		CEN.002 - Usuário autenticado com permissão em uma Base EAB EATG solicita a opção "Nova PT"	Verificar mascara de PT para gerencia executiva GE e area executora com 1 nivel
			Exibição completa das colunas
			Verificar mascara de PT para gerencia executiva AB e area executora com 1 nivel
CEN-003 - Usuário deseja recuperar	Link para Detalhes da PT		

		uma Lista de Programas de Trabalho usando um critério de seleção.	Exibição de PT para usuário Administrador
			Exibição de PT para usuário Não Administrador
UC. 144 - Editar Proposta de Trabalho	3	Usuário Petrobrás autenticado com permissão de “administrador” pelo Project Builder	Conferir se Guias Gestão e Planejamento ficam disponíveis para o usuário após ele selecionar uma PT
	3	Usuário Petrobrás autenticado com permissão diferente de “administrador” pelo Project Builder, com permissão “Incluir Projetos” e já tendo criado uma PT	Conferir se a Guia Gestão fica disponível para o usuário; e se a Guia Planejamento está indisponível, após ele selecionar uma PT
	3	Usuário Petrobrás autenticado com permissão diferente de “administrador” pelo Project Builder, com permissão “Incluir Projetos” e sem ter criado uma PT	Conferir se a opção Listar PT do menu principal não é exibida
	3	Usuário Petrobrás autenticado com permissão diferente de “administrador” pelo Project Builder, sem a permissão “Incluir Projetos”, mas tendo sido cadastrado como Planejador da PT	Conferir se a Guia Gestão fica disponível para o usuário; e se a Guia Planejamento está indisponível, após ele selecionar uma PT
	3	Usuário Petrobrás autenticado com permissão diferente de “administrador” pelo Project Builder, sem a permissão “Incluir Projetos”, mas tendo sido cadastrado como Coordenador da PT	Conferir se a Guia Gestão fica indisponível para o usuário; e se a Guia Planejamento está disponível, após ele selecionar uma PT
UC. 145 - Detalhar Proposta de Trabalho	3	Usuário Petrobrás autenticado com permissão de “administrador” pelo Project Builder	Conferir se Guias Gestão e Planejamento ficam disponíveis para o usuário
	3	Usuário Petrobrás autenticado com permissão diferente de “administrador” pelo Project Builder, mas com permissão “Incluir Projetos”	Conferir se Guias Gestão e Planejamento ficam disponíveis para o usuário
	3	Usuário Petrobrás autenticado com permissão diferente de “administrador” pelo Project Builder, mas sem a permissão “Incluir Projetos”	Conferir se Guias Gestão e Planejamento ficam disponíveis para o usuário
UC. 146 - Excluir Proposta de Trabalho	1	CEN-001 - Usuário autenticado com permissão em uma Base EAB EATG solicita ícone para remoção	Exibir ícone “Remover PT” usuário Planejador
			Exibir ícone “Remover PT” usuário Administrador
			Ausência do ícone “Remover PT” para usuário Coordenador
		CEN-002 – PT com status correspondente para permitir remoção	Permitir remoção da PT com status diferente a

			<p>“Emitida” ou “Aprovada”</p> <p>NÃO Permitir remoção da PT com status “Emitida” ou “Aprovada”</p>
UC. 147 - Importar dados do MS Excel	3	CEN-001 - Usuário autenticado em uma Base EAB EATG solicita a opção "Importar Lista de Documentos"	Opção para Importar Lista de Documentos é exibida no menu Petrobrás
UC. 148 - Gerar Relatórios da PT	2	CEN-001 - Usuário autenticado em uma Base EAB EATG solicita ícones para gerar relatórios	<p>Gerar relatório em PDF com sucesso</p> <p>Exibir Formatação do relatório conforme o modelo fornecido</p> <p>Exibir campos conforme informações da PT em questão</p> <p>Exibir ícones para usuários Planejador, coordenador e Administrador</p> <p>Exibir formatação e conteúdo no relatório da PT DIFERENTE, dependendo da área executora</p>
UC. 149 - Visualizar guia Gestão	3	CEN.001 - Usuário autenticado com permissão em uma Base EAB EATG clica na aba "Gestão"	Opção para Importar Lista de Documentos é exibida no menu Petrobrás
UC. 150 - Visualizar guia Declaração de Escopo	2	CEN.005 - Usuário autenticado em uma Base EAB EATG, e com permissão para Inclusão ou Edição da PT, solicita a opção "Declaração de Escopo"	Conferir se a Guia fica disponível para o usuário
UC. 151 - Visualizar guia Documentação	2	CEN.001 - Usuário autenticado com permissão em uma Base EAB EATG solicita a guia "Documentação"	Conferir se a Guia fica disponível para o usuário
UC. 152 - Visualizar guia Fiscalização de Projetos	2	CEN.006 - Usuário autenticado em uma Base EAB EATG, e com permissão para Inclusão ou Edição da PT, solicita subaba "Fiscalização de Projetos"	Conferir se a Guia fica disponível para o usuário
UC. 153 - Visualizar guia Serviços e Objetos de Custo	2	CEN-001 - Usuário entra com mais de uma viagem nacional por área executora	Conferir se a Guia fica disponível para o usuário
UC. 154 - Visualizar guia Histórico	1	CEN.001 - Usuário autenticado com permissão em uma Base EAB EATG solicita a opção "Histórico"	<p>Exibição da guia “Histórico”.</p> <p>Exibição da coluna “Revisão”.</p> <p>Exibição da coluna “Data”.</p> <p>Exibição da coluna “Descrição”.</p> <p>Exibição dos campos da guia “Histórico” para edição.</p> <p>Campo “Descrição” com</p>

			status em "Emitida"
			Campo "Descrição" com status em "Aprovada"
UC. 155 - Visualizar topo das guias	1	CEN.001 - Usuário autenticado em uma Base EAB EATG, e com permissão para Inclusão ou Edição da PT, solicita a opção "Lista de PT"	Conferir se o topo da pagina fica disponível para o usuário
UC.157 - Visualizar guia Coordenação	2	CEN.001 - Usuário autenticado com permissão em base EAB EATG solicita guia Coordenação	Exibir sub-guia "Coordenador" com usuário "coordenador"
			Exibir sub-guia "Coordenação" com usuário "planejador"
			Exibir sub-guia "Coordenação" com usuário "administrador"
			Campos não editáveis somente conferencia, visualização.
			Exibir cálculos corretamente referente a cada campo da sub-guia
UC.158 - Listar arquivos excel	3	CEN-001 - Usuário autenticado em uma Base EAB EATG solicita a opção "Tabela de Hh e serviços"	Página com tabela com lista de arquivos enviados
			Ícone para download na primeira coluna da Lista de arquivos
			Exibir ordem da tabela de lista de arquivos
			Link na coluna "Ver Log" exibir página "Log de importação de dados tabela Hh e serviços."

APÊNDICE H – Formulários de Avaliação do Estudo de Caso

Neste apêndice são apresentados os formulários de avaliação sobre as mudanças do Processo e sobre o Editor de Casos de Testes utilizado no estudo de caso na organização.

H1 – Formulário de Avaliação das Mudanças no Processo.

Formulário de avaliação Mudanças Processo de Teste

Perfil dos Participantes: Equipe SEPG e demais stakeholders.

O objetivo deste questionário é coletar suas impressões e experiências a partir das mudanças realizadas nas atividades de teste do processo padrão de desenvolvimento de software da organização, e do uso do Editor de Caso de Testes. As respostas registradas serão analisadas e utilizadas como subsídio para a evolução do processo.

Neste sentido, pedimos sua colaboração respondendo as perguntas da forma mais isenta e natural possível, e complementando-as com as observações que se fizerem necessárias.

Percepção sobre mudanças no processo:

- 1) A inserção da atividade “Caracterizar Testes e Planejar Estratégia” na fase de “Especificação e Planejamento do Projeto”
 - (a) Não implicou em qualquer benefício para as atividades de Teste.
 - (b) Dificultou o andamento das atividades de Teste.
 - (c) Facilitou o andamento das atividades de Teste.

Por quê? _____

- 2) A antecipação da atividade “Planejar Testes” para ser executada concorrentemente às atividades de “Preparação da Integração do Produto” e “Especificação Técnica” na fase de “Análise, Projeto e Construção do Software”...
 - (a) Não implicou em qualquer benefício para as atividades de Teste.
 - (b) Dificultou o andamento das atividades de Teste.
 - (c) Facilitou o andamento das atividades de Teste.

Por quê? _____

- 3) A inserção da atividade “Projetar testes do Produto” antes da atividade atual “Planejar os testes do Produto” na fase de “Análise, Projeto e Construção do Software”. ..

- (a) Não implicou em qualquer benefício para as atividades de Teste.
- (b) Dificultou o andamento das atividades de Teste.
- (c) Facilitou o andamento das atividades de Teste.

Por quê? _____

4) A proposição de um template para especificação dos procedimentos, resultados esperados, pré-condições e Casos de Testes...

- (b) Não trouxe qualquer benefício para as atividades de Teste.
- (c) Dificultou o andamento das atividades de Teste.
- (d) Facilitou o andamento das atividades de Teste.

Por quê? _____

5) Qual sua opinião geral sobre os resultados obtidos no uso do Editor de Casos de Teste?

6) Em alguma das atividades do processo de testes (mesmo aquelas não diretamente afetadas pelas mudanças) você percebeu ganho de agilidade? Se sim, qual (is)?

H2 – Formulário de Avaliação do Editor de Casos de Testes

Formulário de Avaliação do Editor de Casos de Testes

Perfil dos Participantes: Analista de testes, Gerente do Projeto e Arquiteto.

Quanto à adoção de uma ferramenta para edição dos Casos de Testes.

1) Indique sua percepção sobre o impacto do uso do Editor de Casos de Testes para a criação dos seguintes artefatos:

Cenários

- (d) Não houve diferença.
- (e) Dificultou.

(f) Facilitou.

Por quê? _____

Casos de Testes

- (a) Não houve diferença.
- (b) Dificultou.
- (c) Facilitou.

Por quê? _____

Procedimentos

- (a) Não houve diferença.
- (b) Dificultou.
- (c) Facilitou.

Por quê? _____

Pré-Condições

- (a) Não houve diferença.
- (b) Dificultou.
- (c) Facilitou.

Por quê? _____

Resultados Esperados

- (a) Não houve diferença.
- (b) Dificultou.
- (c) Facilitou.

Por quê? _____

Variáveis do DataPool

- (a) Não houve diferença.
- (b) Dificultou.
- (c) Facilitou.

Por quê? _____

ACL (Componentes)

- (a) Não houve diferença.
- (b) Dificultou.
- (c) Facilitou.

Por quê? _____

2) Indique sua percepção sobre o impacto do uso do Editor de Casos de Testes para a **manutenção** dos seguintes artefatos:

Cenários

- (g) Não houve diferença.
- (h) Dificultou.
- (i) Facilitou.

Por quê? _____

Casos de Testes

- (d) Não houve diferença.
- (e) Dificultou.
- (f) Facilitou.

Por quê? _____

Procedimentos

- (d) Não houve diferença.
- (e) Dificultou.
- (f) Facilitou.

Por quê? _____

Pré-Condições

- (d) Não houve diferença.
- (e) Dificultou.
- (f) Facilitou.

Por quê? _____

Resultados Esperados

- (d) Não houve diferença.
- (e) Dificultou.
- (f) Facilitou.

Por que _____

Variáveis do DataPool

- (d) Não houve diferença.
- (e) Dificultou.
- (f) Facilitou.

Por quê? _____

ACL (Componentes)

- (d) Não houve diferença.
- (e) Dificultou.
- (f) Facilitou.

Por quê? _____

3) Quanto à capacidade de busca e visualização dos Casos de Testes, o uso da ferramenta:

- não mudou minha capacidade de recuperar os artefatos de testes.
- dificultou a recuperação e visibilidade dos artefatos.
- melhorou a visibilidade dos artefatos porém a capacidade de recuperação de procedimentos e seu posterior reuso permanece a mesma.
- melhorou a visibilidade dos artefatos e facilitou a recuperação de procedimentos e seu posterior reuso.

4) Qual sua opinião geral sobre as mudanças no processo, incluindo o uso do Editor?

5) Em alguma das atividades do processo (mesmo aquelas onde as mudanças não ocorreram diretamente) foi percebido ganho de agilidade? Se sim, qual (is)?

APÊNDICE I – Formulários de Consentimento de participação no estudo

Neste apêndice são apresentados o formulário de consentimento preenchido pelos participantes do estudo de caso realizado na organização.

I1 – Formulário de Consentimento dos Participantes

Estudo sobre a adoção de Práticas de Agilidade em Atividades de Projeto e Construção de Casos de Testes

Eu declaro ter mais de 18 anos de idade e que concordo em participar nos estudos conduzidos pelo pesquisador **Ciro Grippi Barbosa Lima**, como parte das atividades do curso de Pós Graduação em Engenharia de Software da UFRJ.

Estes estudos visam melhor compreender a adoção de práticas de agilidade em atividade de um processo de testes, tentando entender sob que condições estas características podem ser introduzidas nas atividades.

PROCEDIMENTO

Será disposto um software editor de casos de testes e um conjunto de diretrizes será passado para os analistas e testadores de forma a facilitar seu uso. Um template de Caso de Teste será proposto para ser utilizado. Esta ferramenta e o template serão utilizados na execução das atividades de processo de teste que fazem parte do desenvolvimento de projetos de software da organização que trabalho.

O pesquisador conduzira o estudo consistindo da coleta, análise e relato dos dados obtidos nas etapas de projeto, construção e execução dos Casos de Testes. Eu entendo que não tenho obrigação alguma em contribuir com informação sobre meu desempenho nestas atividades, e que posso solicitar a retirada de meus resultados a qualquer momento e sem qualquer penalidade ou prejuízo. Eu entendo que não existira nenhum crédito ou benefício extra por participar deste estudo. Eu entendo também que quando os dados forem coletados e analisados, meu nome será removido dos dados e que este não será utilizado em nenhum momento durante a análise ou quando os resultados forem apresentados.

CONFIDENCIALIDADE

Toda informação coletada neste estudo é confidencial, e meu nome não será identificado em momento algum. Da mesma forma, me comprometo a não comunicar os meus resultados enquanto não terminar o estudo, bem como manter sigilo das técnicas e documentos apresentados e que fazem parte deste estudo.

BENEFÍCIOS, E LIBERDADE DE DESISTÊNCIA.

Eu entendo que os benefícios que receberei deste estudo são limitados ao aprendizado do material que é distribuído e ensinado, independente de participar ou não deste estudo, mas que os pesquisadores esperam aprender mais sobre a adoção

de características de agilidade em processos de testes, e os benefícios trazidos por este estudo para o contexto da Engenharia de Software.

Eu entendo que sou livre para realizar perguntas a qualquer momento ou solicitar que qualquer informação relacionada a minha pessoa não seja incluída no estudo. Eu entendo que participo de livre e espontânea vontade com o único intuito de contribuir para o avanço e desenvolvimento de técnicas e processos para Engenharia de Software.

PROFESSOR RESPONSÁVEL Prof. Guilherme Horta Travassos
Programa de Engenharia de Sistemas e Computação
COPPE/UFRJ

APÊNDICE J – Formulários de Caracterização dos Participantes

Neste apêndice são apresentados os formulários de caracterização dos participantes do estudo de caso realizado na organização.

J1 – Formulário de Caracterização dos Participantes

Nome:

Formação Geral

Qual é sua experiência anterior com desenvolvimento de software na prática? (marque aqueles itens que melhor se aplicam)

- nunca desenvolvi software.
- tenho desenvolvido software para uso próprio.
- tenho desenvolvido software como parte de uma equipe, relacionado a um curso.
- tenho desenvolvido software como parte de uma equipe, na indústria.

Qual é sua experiência anterior com Projeto de Casos de Teste na prática? (marque aqueles itens que melhor se aplicam)

- nunca projetei/construí Casos de Teste.
- já projetei/construí Casos de Teste em projetos no passado.
- tenho projetado Casos de Teste como parte de uma equipe, relacionado a um curso.
- tenho projetado Casos de Teste como parte de uma equipe, na indústria.

Qual é sua experiência anterior com atividades de Execução de Casos de Teste na prática? (marque aqueles itens que melhor se aplicam)

- nunca executei Casos de Teste.
- já trabalhei como testador em Casos de Teste em projetos no passado.
- tenho atuado como testador, fazendo parte de uma equipe, relacionada a um curso.
- tenho atuado como testador, fazendo parte de uma equipe, na indústria.

Por favor, explique sua resposta. Inclua o número de semestres ou número de anos de experiência relevante em testes (E.g. “Eu trabalhei por X anos como analista de Teste, ou Eu trabalhei por X anos como testador, ou Eu trabalhei por X anos conduzindo projetos de Teste”) (Caso tenha experiência com testes, relate o seu tempo de experiência e tipos de testes)

Experiência em Testes de Software

Por favor, indique o grau de sua experiência nesta seção seguindo a escala de 5 pontos abaixo:

1 = nenhum

2 = estudei em aula ou em livro

3 = pratiquei em 1 projeto em sala de aula

4 = usei em 1 projeto na indústria

5 = usei em vários projetos na indústria

Resp:

Experiência com Técnicas de Teste

- Experiência com Classe por Equivalência 1 2 3 4 5
Resp:
- Experiência com Análise de Valor-Limite 1 2 3 4 5
Resp:
- Experiência com Grafo de Causa-Efeito 1 2 3 4 5
Resp:

Experiência em Projeto

- Experiência escrevendo casos de uso 1 2 3 4 5
Resp:
- Experiência desenvolvendo Casos de Testes 1 2 3 4 5
Resp:
- Experiência executando Casos de Testes 1 2 3 4 5
Resp:

Outras Experiências

- Experiência em desenvolver Casos de testes a partir de requisitos e casos de uso 1 2 3 4 5
Resp:
- Experiência com testes de integração de software? 1 2 3 4 5
Resp:
- Experiência com testes de unidade? 1 2 3 4 5
Resp:
- Experiência com testes funcionais? 1 2 3 4 5
Resp:
- Experiência com testes de aceitação? 1 2 3 4 5
Resp:

Experiência em Contextos Diferentes

Nós usaremos esta seção para compreender quão familiar voce está com vários sistemas que poderão ser utilizados como exemplos ou para exercícios durante o curso.

Por favor, indique o grau de experiência nesta seção seguindo a escala de 3 pontos abaixo:

1 = Eu não tenho familiaridade com a área. Eu nunca fiz isto.

3 = Eu utilizo isto algumas vezes, mas não sou um especialista.

5 = Eu sou muito familiar com esta área. Eu me sentiria confortável fazendo isto.

Quanto voce sabe sobre...

- Utilizar Ferramentas de Projeto de Testes? 1 3 5
Resp:
- Utilizar ferramentas p/ acompanhamento das atividades de Testes? 1 3 5
Resp:
- Utilizar os serviços de uma vídeo-locadora? 1 3 5
Resp:

APENDICE K - EditorCT: Ferramenta para Projeto e Construção de Casos de Testes

Neste apêndice serão descritos os desdobramentos da proposição de um template para apoiar a inserção de agilidade e contribuir para o reuso dos artefatos de software, através do desenvolvimento de uma ferramenta de edição de Casos de Testes que visa facilitar o projeto e reaproveitamento de elementos de Casos de Testes desenvolvidos durante a evolução de um determinado projeto de software.

K.1 Introdução

Neste apêndice estão dispostos os requisitos derivados do trabalho de pesquisa na literatura técnica e das observações do contexto das atividades de teste em uma organização selecionada para participar do estudo.

Na etapa de projeto foi desenvolvido um Wire-Frame bem como diagramas de classes e Casos de Uso. Foi utilizado o *framework Ruby On Rails* na etapa de construção da ferramenta.

K.2 Editor de Casos de Teste

K.2.1 Entidades e Requisitos do Editor de Casos de Teste

K.2.1.1 Entidades do Modelo

As entidades que farão parte deste modelo estão listadas na sequência:

- Projeto: Um projeto é um empreendimento não repetitivo, caracterizado por uma sequência clara e lógica de eventos com início, meio e fim, que se destina a atingir um objetivo claro e definido, sendo conduzido por pessoas dentro de parâmetros predefinidos de tempo, custo, recursos envolvidos e qualidade (VARGAS, 2006). Para o escopo destas ferramentas serão considerados apenas projetos de software.
- Release - Representa uma versão de um determinado produto podendo configurar uma evolução de um mesmo produto ou ainda uma especialização de um conjunto de produtos de um mesmo domínio a ser destinada exclusivamente a um determinado cliente.

- Requisitos: De acordo com definição descrita no guia MPS.BR parte 1 (SOFTEX, 2011) um requisito representa a capacidade requerida pelo usuário que deve ser encontrada ou possuída por um determinado produto, ou componente de produto para resolver um problema ou alcançar um objetivo ou para satisfazer a um contrato, a um padrão, a uma especificação ou a outros documentos formalmente impostos. Um requisito pode ser classificado em requisitos funcionais e não funcionais. Os requisitos funcionais expressam o comportamento de um software. As informações de entrada, o processamento e a saída emitida por uma funcionalidade são informações necessárias para especificar o requisito do referido grupo. Já os requisitos não funcionais mapeiam os aspectos qualitativos de um software, por exemplo: desempenho; segurança; perspectiva do usuário; comunicabilidade; usabilidade e portabilidade (FABRI, 2010).
- Casos de Usos (UC): “Um caso de uso é uma sequência especial de operações, realizadas por um usuário e um sistema através do estabelecimento de um diálogo. A operação é realizada pela o usuário ou o sistema e é uma resposta a um determinado estímulo” (JACOBSON, 1987). Um Caso de Uso apresenta um conjunto de caminhos de execução cada um representando um fluxo alternativo de execução.
- Cenários: Um cenário de um determinado Caso de Uso pode representar o fluxo principal de execução, um dos caminhos alternativos, ou ainda situações de exceção.
- Casos de Testes: Um caso de testes é um par de resultados esperados e de entradas, aonde as entradas são dados a serem enviados para o elemento do *software* sob teste, e resultado esperado é uma descrição da saída que o *software* deve exibir para a entrada associada (MCGREGOR ET SYKES, 2001).
- Procedimento de Casos de Teste: Conjunto de passos necessários para executar um conjunto de Casos de Testes.
- Planos: O Gerente de Testes deverá definir as estratégias de Testes a serem aplicadas – unidade, integração, sistema, aceitação, re-teste (regressão e *smoke test*) e instalação. Sendo que para cada um destas estratégias deverá ser proposto um plano, composto de uma sequência de casos de Testes. Nos testes não funcionais deverá ser considerado o tipo dos testes (desempenho, carga, stress e outros) para a composição do plano e seus respectivos Casos de testes.

- **Passos:** Um passo corresponde a uma ação a ser executada pelo testador. Através da execução de um conjunto de passos (procedimentos) o testador será capaz de executar o Caso de Testes definido pelo analista.
- **Pré-condições:** Um conjunto de condições a serem atendidas para que os testes possam ser executados. Um Caso de Testes pode herdar pré-condições do Cenário associado. Na ferramenta proposta um bloco de pré-condições irá conter a sequência de passos necessários para a verificação das pré-condições de cada caso de Teste.
- **DataPool :** Uma lista de dados de entrada a serem repassados para um determinado Caso de Teste.
- **Resultados Esperados:** O estado resultante ou as condições esperadas na execução de um determinado Caso de teste. Na ferramenta proposta um bloco de resultados esperados irá conter a sequência de passos necessários para a verificação dos resultados esperados definidos pelo analista.
- **Suíte de Testes:** Os Casos de testes podem ser agrupados em uma coleção a partir de uma classificação a ser definida pelo gerente de testes com o objetivo de organizar sua execução e auxiliar na determinação de quais testes deverão ser reexecutados em uma estratégia de regressão. Apesar de o modelo considerar este elemento, o estudo realizado estará utilizando o agrupamento dos Casos de testes por cenários.
- **ACL:** Acrônimo de *Action List*. Uma *Action List* é uma ação a ser associada opcionalmente a um passo de um procedimento de Caso de Teste de uma rotina de verificação de pré-condição, ou de uma rotina de verificação de um resultado esperado. Cada *Action List* definida na ferramenta será armazenada na base de dados, para permitir sua reutilização pelos vários procedimentos definidos a posteriori. Uma *Action List* pode conter de 0 a N *Action Lists*.
- **Parâmetros:** Uma ACL pode estar associada a parâmetros. Por exemplo, em um sistema de Gerenciamento de Projetos, uma ACL responsável pelo cadastramento de um projeto deverá fazer referencia a um parâmetro, capaz de conter o nome do projeto a ser cadastrado. Cada parâmetro possui um tipo associado, por exemplo, *string*, numérico etc.
- **Comandos SDK:** Uma ACL poderá estar associada a um comando SDK de um motor de execução de testes automatizados. Neste caso esta ACL estará fazendo referencia a um comando da API da ferramenta de testes automatizado. Este elemento foi incluído para implementações futuras.

- Variável: Um *DataPool* irá conter uma lista de variáveis de entradas a serem informadas pelo testador, na etapa de execução dos Casos de Teste. Estas variáveis, assim como os parâmetros possuem tipos associados.
- Execução: As informações de execução de um caso de teste, como autor, data, *status* da execução deverão ser armazenadas pela ferramenta para acompanhamento.
- Usuários: Cada usuário da ferramenta possuirá um *login* e senha, sendo seu nome cadastrado na base de dados. Um usuário irá possuir um único perfil para cada release de projeto em que ele venha a participar da equipe de testes, sendo possível ser associado a um projeto com um perfil por exemplo de “Analista de Teste” e em outro projeto com o perfil de Testador.
- Perfil: Existem quatro perfis cadastrados previamente na ferramenta: Testador, Analista de Testes, Gerente e arquiteto. O testador é o responsável pela execução dos testes e pelo registro do resultado destas execuções na ferramenta. O Analista de testes responsável pelo cadastramento dos artefatos de desenvolvimento e pelo projeto dos casos de testes, procedimentos, pré-condições, resultados esperados, e outros elementos de teste. O arquiteto é responsável pelo cadastramento das ACLS. O Gerente de testes acompanha as tarefas de projeto e execução e dos Casos de Testes.
- Regras de negócios: Para cada projeto existirá um conjunto de regras específicas do domínio de problema associado. Estas regras elicítadas conjuntamente com os *stakeholders* deverão ser registradas, pois estas serão utilizadas como fonte para estabelecimento das pré-condições, entradas e resultados esperados dos Casos de Testes.

Na etapa de projeto foi criado um diagrama de classes contendo as entidades elicítadas e definidas nesta etapa.

Como a organização já utilizava uma ferramenta de software livre, além dos requisitos associados à inserção da prática de agilidade foi também considerado um projeto de tela (wire-frame) que aproveitasse algumas boas características de navegabilidade presentes nesta ferramenta utilizada pela organização. O objetivo com esta estratégia foi tentar mitigar eventuais dificuldades de aprendizado do uso da ferramenta na etapa de execução do estudo de caso.

Apesar do foco do estudo estar centrado nas atividades de projeto de testes foi necessário implementar funcionalidades que permitissem a emissão de relatórios de incidentes de teste, e a geração do plano contendo todos os Casos de Testes

desenvolvidos para que fossem anexados à nova versão do Plano de testes contemplando as informações geradas na etapa de caracterização dos testes.

Na figura K-1 é exibido o modelo de Classes a ser utilizado no desenvolvimento da ferramenta.

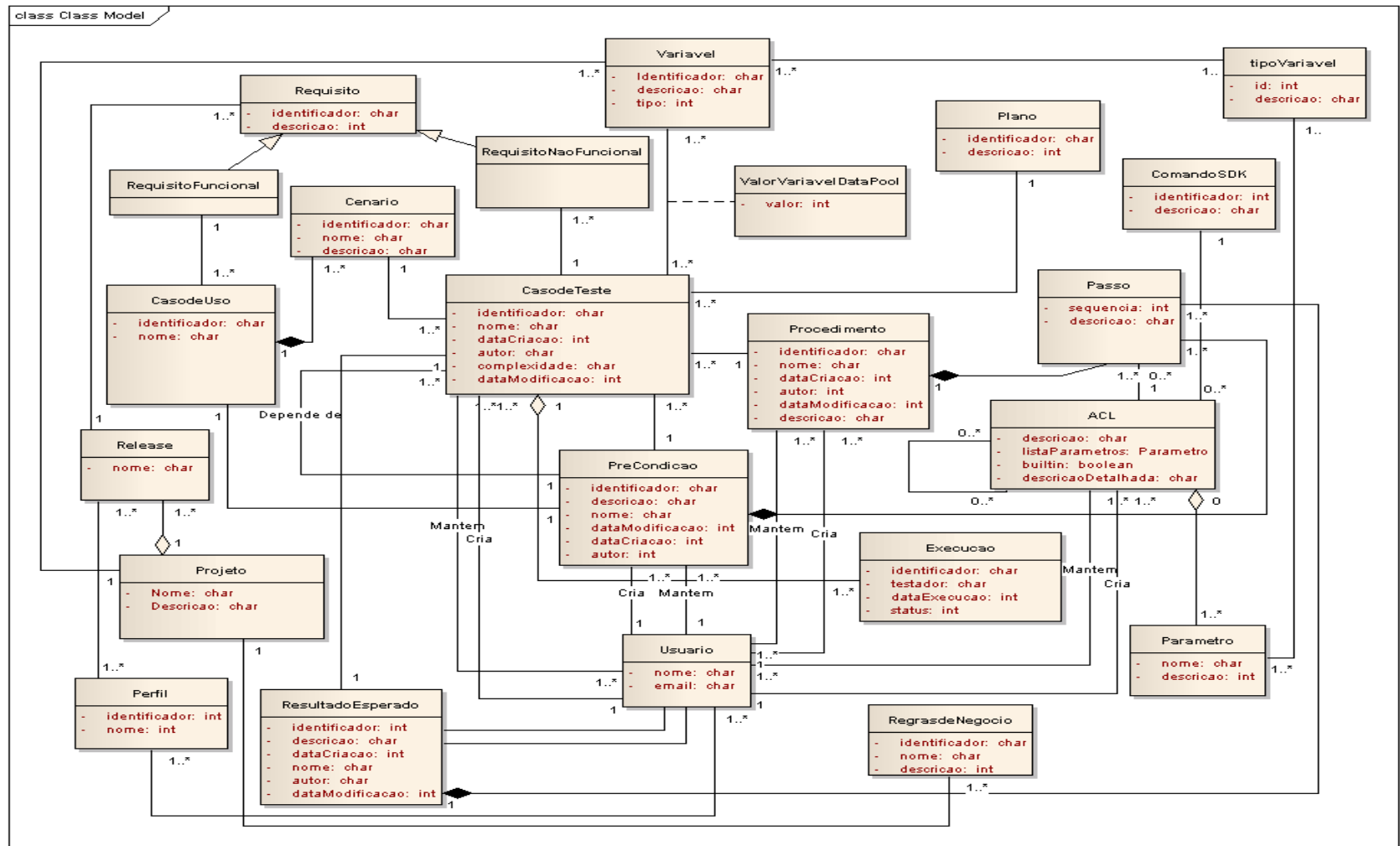


Figura K-1 - Diagrama de Classes da Ferramenta de Edição de Casos de Testes

K.2.1.2 Requisitos do Editor

Além da elicitación e análise dos requisitos funcionais foram elencados os requisitos não funcionais da ferramenta, sendo priorizados aspectos de usabilidade e navegabilidade para prover facilidade de visualização da rastreabilidade dos requisitos, casos de uso e cenários cadastrados aos casos de Teste, como também rastreabilidade entre procedimentos, pré-condições, entradas, resultados esperados a estes mesmos artefatos, em um esquema bidirecional. Para atender estas premissas, na implementação das telas foi considerado o uso de *auto complete*, mapas e outros recursos.

K.2.1.2.1 Requisitos Funcionais

Requisito Funcional 1:

Os usuários deverão ser cadastrados e associados a um perfil de utilização por release de projeto, sendo que a cada um destes perfis está associado um conjunto de permissões associadas às funcionalidades. São quatro perfis disponíveis para associação: (a) Gerente de Testes, (b) Analista de Testes, (c) Testador (d) Arquiteto.

Requisito Funcional 2:

Os gerentes e analistas de Teste deverão ser capazes de cadastrar e manter os projetos cujos artefatos de testes serão cadastrados no sistema. Os testadores, e os arquitetos responsáveis pela criação dos componentes (*Action List*) serão capazes de listar e consultar as informações relativas a estes projetos.

Requisito Funcional 3:

Os gerentes e analistas de Teste deverão ser capazes de cadastrar e manter os releases que estarão associados a cada projeto cadastrado no sistema. Os testadores, e os arquitetos de componentes serão capazes de listar e consultar as informações relativas aos releases.

Requisito Funcional 4:

Os gerentes e analistas de Teste deverão ser capazes de cadastrar e manter os requisitos funcionais e não funcionais a cada projeto cadastrado no sistema. Os testadores, e os arquitetos de componentes serão capazes de listar e consultar as informações relativas aos requisitos.

Requisito Funcional 5:

Os gerentes e analistas de Teste deverão ser capazes de cadastrar e manter os Casos de Uso e associa-los aos requisitos funcionais cadastrados no sistema. Os

testadores, e os arquitetos de componentes serão capazes de listar e consultar as informações relativas aos Casos de Uso.

Requisito Funcional 6:

Os gerentes e analistas de Teste deverão ser capazes de cadastrar e manter cada um dos cenários e associa-los a um determinado Caso de Uso cadastrado previamente no sistema. Os testadores, e os arquitetos dos componentes serão capazes de listar e consultar as informações relativas aos cenários.

Requisito Funcional 7:

Os gerentes, testadores e analistas de Teste deverão ser capazes de cadastrar e manter os planos de Testes para os releases criados previamente no sistema. Os arquitetos de componentes serão capazes de listar e consultar as informações relativas aos cenários.

Requisito Funcional 8:

Os gerentes e analistas de Teste deverão ser capazes de cadastrar e manter as Regras de Negócios para os projetos cadastrados previamente no sistema. Os testadores, e os arquitetos de componentes serão capazes de listar e consultar as informações relativas às regras de negócios.

Requisito Funcional 9:

O analista e o gerente de testes deverá ser capaz de cadastrar e manter os casos de testes e estabelecer uma associação destes Casos de Testes aos cenários criados previamente, sendo que cada Caso de Testes possuirá um conjunto de dados de entrada (*data pool*), pré-condições, resultados esperados e procedimentos. Os demais perfis serão capazes de listar o conteúdo dos casos de testes e seus itens associados.

Requisito Funcional 10:

Para cada Caso de Teste criado deverá ser informado um plano de teste cadastrado previamente.

Requisito Funcional 11:

O analista de teste deverá ser capaz de cadastrar e manter os procedimentos e pré-condições a serem associados aos Casos de Testes cadastrados no sistema.

Requisito Funcional 12:

O analista de teste deverá ser capaz de cadastrar e manter os resultados esperados e o *data pool* de dados de entrada a serem associados aos Casos de Testes cadastrados no sistema.

Requisito Funcional 13:

O arquiteto deverá ser capaz de cadastrar e manter componentes (*ActionList*) os quais poderão ser associados a passos de cada procedimento de teste, verificação

de resultado esperado ou pré-condição. Opcionalmente poderá ser concedida ao analista de teste estas permissões, dependendo das características da equipe e de decisão do gerente de testes.

Requisito Funcional 14:

Os procedimentos, pré-condições, *data pool* e resultados esperados são associados na etapa de sua criação a um determinado projeto podendo ser reutilizados pelos casos de Testes de releases de um mesmo projeto.

Requisito Funcional 15:

Deverá ser permitida aos usuários da ferramenta a recuperação dos Casos de testes de um determinado release, especificando como filtro um Plano de Testes selecionado previamente.

Requisito Funcional 16:

No cadastramento de Casos de Testes o analista de teste opcionalmente poderá especificar o nível de complexidade do Caso de Teste, a partir de taxinomia proposta.

Requisito Funcional 17:

Apenas usuários cadastrados poderão acessar o sistema.

Requisito Funcional 18:

Deverá ser possível aos usuários do sistema recuperar todos os Casos de Testes associados a um determinado procedimento.

Requisito Funcional 19:

Na funcionalidade de edição de procedimento de caso de testes deverá ser oferecida ao analista de testes a capacidade de visualizar o conteúdo de cada *ActionList* que faz parte do procedimento. Esta mesma possibilidade deverá ser oferecida nas funções de resultados esperados, edição de pré-condições.

Requisito Funcional 20:

Deverá ser assegurada a rastreabilidade bidirecional entre Projeto, Release, Requisitos, Caso de Usos, Cenários e Casos de Testes.

Requisito Funcional 21:

Os usuários poderão a qualquer momento gerar um relatório contendo os Casos de Testes desenvolvidos para um determinado release de projeto passando como filtro um plano de Testes cadastrado previamente.

Requisito Funcional 22:

O testador deverá ser capaz de registrar a execução de cada Caso de Teste cadastrado na ferramenta informando o resultado da execução (“Falha”, “Passou”) e ainda sendo capaz de suspender sua execução associando o resultado “Suspendido”.

Requisito Funcional 23:

Os usuários poderão a qualquer momento gerar um relatório de incidentes das atividades de execução dos CTs, sendo listados todos os Casos de Testes associados ao plano selecionado e para cada um deles exibido o estado de execução de cada um destes artefatos (“Não executado”, “Falhou”, “Passou”, “Suspenso”).

K.2.1.2.2 Requisitos Não-Funcionais

Usabilidade

Requisito Não Funcional 1:

A ferramenta deverá implementar recurso capaz de facilitar a navegabilidade entre os elementos que formam a matriz de rastreabilidade composta por requisitos, Casos de Uso, Cenários, Casos de Testes, procedimentos, pré-condições, resultados esperados e *data pool*. Esta hierarquia deverá ser refletida de forma a facilitar seu entendimento, por exemplo, através da exibição de uma árvore contendo cada um destes elementos.

Requisito Não Funcional 2:

A ferramenta deverá facilitar a navegabilidade entre os componentes que formam uma *Action Lists* exibindo o relacionamento entre os vários subcomponentes, por exemplo, através da exibição de uma árvore contendo cada uma das ACLS.

Requisito Não Funcional 3:

Na etapa de criação dos Casos de Testes deverá ser implementado recurso que facilite a visualização dos procedimentos, pré-condições, resultados esperados cadastrados previamente na ferramenta. Deverá ser estudada a aplicabilidade do recurso de *auto complete*, assim como qualquer outro recurso que facilite a leitura e recuperação dos elementos cadastrados previamente e que irão compor os casos de Testes.

Requisito Não Funcional 4:

Na criação dos procedimentos deverá ser implementado um recurso que facilite a visualização dos componentes (ACLS) que foram cadastrados previamente na ferramenta e que poderão ser associados a cada passo de um determinado procedimento. O objetivo é promover a identificação e reutilização dos procedimentos pelos vários releases de um projeto de desenvolvimento de um software.

K.2.1.2.3 Regras de Negócios

Regra de Negócio RN1:

Um passo pode estar associado a uma lista de ações (ACL).

Regra de Negócio RN2:

Uma ACL (*Action List*) pode estar associada a um conjunto de *Action Lists*.

Regra de Negócio RN3:

As ACLS podem ser decompostas em várias outras ACLS através de uma associação. Esta decomposição pode ser realizada até um nível em que estará sendo descrito, por exemplo, o acesso a um botão de tela, uma solicitação de uma URL, o preenchimento de um campo. Neste nível de detalhamento é possível automatizar sua execução, seja através de um comando de interface (no caso de um testes funcional em modo interativo com o usuário) ou, por exemplo, através de um comando de consulta a uma base de dados para verificação da existência de um registro ou de um valor atribuído a um determinado campo (em um bloco de resultado esperado). Neste caso esta ACL deverá ser categorizada como uma ação que permite sua associação a um comando SDK.

Regra de Negócio RN4:

Os identificadores de requisitos deverão ser únicos para um determinado projeto.

Regra de Negócio RN5:

Os identificadores de cada Caso de Uso deverão ser únicos para cada requisito.

Regra de Negócio RN6:

Os identificadores de cenários deverão ser únicos para um determinado Caso de Uso.

Regra de Negócio RN7:

Os identificadores dos Casos de Testes deverão ser únicos para cada cenário.

Regra de Negócio RN8:

Os identificadores de Procedimentos, Resultados Esperados e Pré-Condições deverão ser únicos para um determinado projeto.

Regra de Negócio RN9:

Na criação de um procedimento e seus respectivos passos é facultado ao analista de testes a associação de cada passo a uma ACL criada previamente.

Regra de Negócio RN10:

Ao solicitar a remoção de uma determinada ACL de um determinado projeto, utilizada por outras ACLS, o sistema deverá informar ao usuário a relação das ACLS que a utilizam solicitando confirmação desta exclusão. Caso o usuário confirme esta remoção, tanto a ACL quanto todos os seus relacionamentos com outras ACLS serão removidos.

Regra de Negócio RN11:

Ao solicitar a remoção de uma determinada ACL de um determinado projeto constituída de uma mais ACLS o sistema excluirá apenas a ACL solicitada não removendo a árvore de ACLS que a compõem. Há não ser que o usuário informe previamente a necessidade de remoção de toda a árvore.

Regra de Negócio RN12:

Ao exibir os parâmetros de um determinado procedimento cujos passos foram associados à ACLS o sistema deverá recuperar a lista de todos os parâmetros de cada componente utilizado.

Regra de Negócio RN13:

Ao associar um membro da equipe a um determinado release de um projeto deverá ser informado o tipo de perfil que este participante estará atuando. Na etapa de instalação do sistema serão criados quatro perfis, a saber: (a) Gerente de Testes, (b) Analista de Testes, (c) Testador (d) Arquiteto.

Regra de Negócio RN14:

O perfil gerente de testes será capaz de acessar as funções de manutenção dos artefatos utilizados na etapa de planejamento e projeto de testes (releases, requisitos, casos de usos, cenários, planos), a função que permite editar um Caso de testes a partir de sua busca em uma árvore, e a função de exibição da árvore contendo os componentes (ACLS), sendo neste caso disponíveis apenas para a leitura. O gerente de testes poderá também acessar o menu de execuções dos CTs.

Regra de Negócio RN15:

O perfil analista de testes será capaz de acessar todas as funções dispostas para o Gerente com exceção das funções relacionadas à execução dos Casos de Testes.

Regra de Negócio RN16:

O perfil testador será capaz de acessar as funções de consulta aos artefatos de projeto de Testes. Possuía permissão de atualização apenas para as funções de registro de execução dos Casos de Testes.

Regra de Negócio RN17:

Os perfis arquiteto e analista de teste serão capazes de criar, editar e excluir componentes.

Regra de Negócio RN18:

Ao relacionar um procedimento de testes a um Caso de Testes deverá ser informado, para cada um de seus parâmetros qual a variável a ser associada do *data pool* de Caso de Teste.

Regra de Negócio RN19:

Na funcionalidade de edição de um determinado procedimento, à medida que o usuário for associando as listas de ações aos passos do procedimento, o sistema deverá acrescentar à lista de parâmetros do procedimento os parâmetros que porventura existem na lista de ações.

Regra de Negócio RN20:

Não será possível apagar um requisito que apresente casos de Usos associados. Não será possível apagar Cenários que possuam Casos de Testes associados.

Regra de Negócio RN21:

Não será possível apagar um Caso de Testes que já tenha tido seu status alterado para executado com sucesso (“Passou”) ou executado com insucesso (“Falhou”). Se o Caso de Testes não teve o seu status alterado, o analista de Testes poderá remover o Caso de Testes da base de dados.

Regra de Negócio RN22:

Não será possível registrar uma data de execução com data anterior a última execução registrada.

Regra de Negócio RN23:

Os procedimentos, pré-condições, resultados esperados, entradas, variáveis e componentes pertencem a um determinado projeto e podem ser compartilhados entre seus releases.

Regra de Negócio RN24:

Os Requisitos, Casos de Uso, Cenários e Casos de Testes pertencem a um determinado release de um determinado projeto.

K.2.2 Diretrizes para a Criação de Casos de Testes

Conjuntamente com as atividades de elicitação dos requisitos para o projeto e desenvolvimento da ferramenta de Edição de Casos de Testes foi realizado levantamento dos artefatos utilizados pela organização no projeto e construção dos Casos de Testes (template do documento de requisitos, matriz de rastreabilidade e gabarito de Casos de Uso).

O template de Casos de Uso utilizado pela organização fica cadastrado na ferramenta *Enterprise Architect* (EA) contemplando a definição dos fluxos alternativos e de exceção, além do fluxo principal. Os analistas ao iniciarem um novo projeto instanciam uma versão de um arquivo padrão EA contendo este e outros templates utilizados em seu processo de Projeto e Construção de Produto (PCP).

Na abordagem utilizada pela organização, os Casos de Teste Funcionais são derivados de cada fluxo definido no Caso de Uso. Sendo os CTs agrupados em Suítes de Teste e o critério para este agrupamento é *ad hoc* dependendo de cada analista de teste envolvido.

Com o objetivo de facilitar a recuperação dos procedimentos e CTs a serem modificados por conta de uma solicitação de requisitos e, considerando que a equipe já possui como prática o desenvolvimento de Casos de Uso e, os respectivos fluxos, será sugerida uma abordagem na qual estes fluxos serão utilizados como um critério de agrupamento de CTs (denominados cenários). O analista de Teste deverá mapear os possíveis Fluxos de Eventos do Caso de Uso (UC) para desenvolver os CTs que irão exercitar estes caminhos, e também deverá verificar se existem alternativas de fluxo que não foram explicitadas no UC. No caso de existir, deverá ser debatida junto à equipe de desenvolvimento e stakeholder a necessidade de inclusão, modificação dos fluxos existentes ou reescrita. Para cada cenário de execução um conjunto de Casos de Testes (1 a N) será projetado.

Cada Caso de Uso poderá estar associado a uma pré-condição. Uma pré-condição é uma condição que deverá ser observada para que o Caso de Uso possa ser executado. A partir das pré-condições dos Casos de Usos deverão ser especificadas as pré-condições dos Casos de Testes. O analista de teste avaliando os passos e pré-condições do Caso de Uso será capaz de determinar as variáveis de entrada necessárias para cada Caso de Teste de cada Cenário estabelecido (Nos valores a serem escolhidos para entrada é aconselhável o uso das técnicas de classe por equivalência, análise valor-limite e grafo).

Da mesma forma, UCs possuem pós-condições que definem o estado do sistema após a execução. Fluxos diferentes associados a distintos cenários poderão apresentar pós-condições distintas. Pós-condições dos Casos de uso serão utilizadas para a definição das pós-condições e resultados esperados de cada Caso de Teste.

Por último deverão ser examinadas as Regras de negócio porque conjuntamente com as pré-condições dos Casos de Uso serão utilizadas na definição da pré-condição dos CTs.

K.2.3 Definição do Wire-Frame

Na definição das telas do Editor de Casos de Testes um *Website Wire frame* foi utilizado para especificar a estrutura de navegação, definir o relacionamento entre as páginas e retratar a arquitetura das informações a serem manipulada pelos usuários da ferramenta.

Nesta etapa o objetivo principal foi definir um conjunto de telas e um esboço de navegação de forma que os elementos que compõem os Casos de Testes – a saber, procedimentos de execução, pré-condições, resultados esperados, e variáveis de entrada – pudessem ser editados separadamente e ao mesmo tempo facilmente associados aos Casos de Teste desenvolvidos pelos analistas, procurando obter visibilidade e recuperação rápida visando o reuso.

Na figura K-2 é mostrado o *wire frame* da tela de edição de Casos de Testes. O objetivo é guiar o analista a separar adequadamente as responsabilidades envolvidas em cada elemento de cada Caso de Teste para que passos de pré-condições não venham a ser codificadas conjuntamente com os passos de procedimentos de execução dos testes e nem tampouco valores de entrada e resultados esperados se misturem conjuntamente com procedimentos.

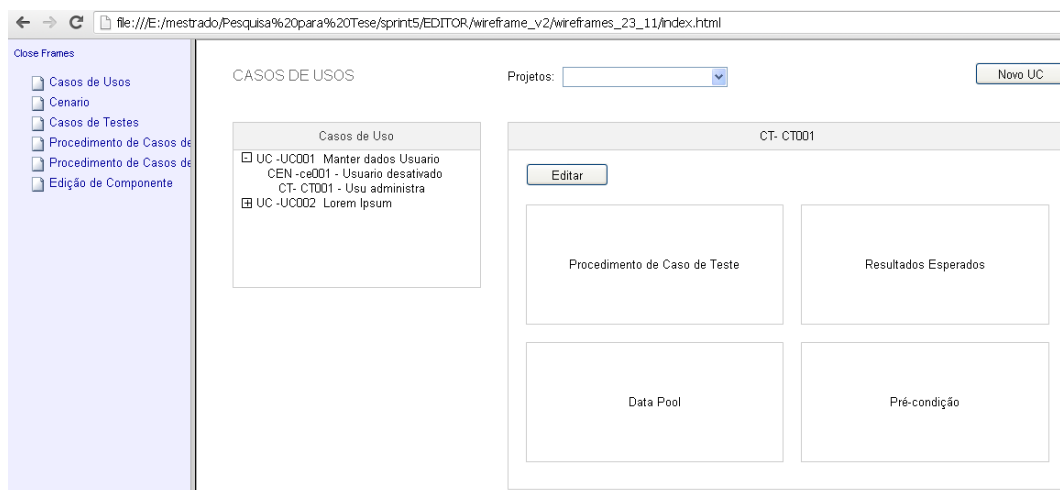


Figura K-2 - *Wire-Frame* para Tela de Consulta a Casos de Testes

Ainda na figura K-2, na parte esquerda da tela de *Wire Frame* é mostrada uma árvore em que os Casos de Teste de cada Caso de Uso podem ser acessados, implementando uma rastreabilidade entre estes elementos - como é disposto em algumas das ferramentas que implementam esta rastreabilidade - como, por exemplo, o *TestLink*. Na etapa da construção das telas esta rastreabilidade foi estendida a outros elementos permitindo a navegação desde o requisito funcional, passando pelo caso de Uso, cenário, Caso de Teste, procedimento de execução, pré-condição, entradas e resultados esperados. Além disso, foi disposta uma árvore no sentido procedimento a Casos de Testes de forma a ser possível determinar visualmente quais Casos de Testes estão associados a que procedimentos, atendendo às necessidades de rastreabilidade bidirecional. Ao clicar no botão editar o analista pode acessar os vários elementos que compõem o CT, criando novas associações e alterações que se fizerem necessárias (Figura K-3).

Close Frames

- Casos de Usos
- Cenário
- Casos de Testes
- Procedimento de Casos de Testes
- Procedimento de Casos de Testes Detalhado
- Edição de Componente

EDIÇÃO DE CASOS DE TESTES Voltar

Caso de Uso: Caso de Teste:

Cenário: Descrição:

Procedimento

Nome: Novo Editar

Lista: Associar

Passos	Componente
1 - Entre com Login Y, senha X	
2 - Criar Projeto X	
3 - Salvar Projeto	

Resultado Esperado

Nome: Novo

Passos	Componente	Edição
1- Verifique se na tabela X foi inserido o reg 10		Editar

Pré-Condição

Nome: Novo

Passos	Componente	Edição
1- Verifique se o campo vol do reg de estoque tem valor 10		Editar

Data Pool

Variável	Valor

Figura K-3 - Wire Frame da Tela de Edição de Casos de Testes

Cada procedimento, rotina de verificação de resultado esperado e rotina de verificação de pré-condição é editada separadamente, permitindo que cada passo de procedimento possa ser associado a um componente criado previamente,

Foi projetada também nesta etapa uma tela de edição de ACLS para que o arquiteto de componentes de testes e/ou o analista de testes possam criar componentes para que posteriormente possam ser associados a passos de cada procedimento de teste definido - seja este procedimento um conjunto de passos de verificação de pré-condição ou de resultados esperados ou de execução de caso de teste. A ideia foi permitir que a edição pudesse ser realizada utilizando ACLS definidas previamente ou ainda criando novas ACLS à medida que o analista perceba a necessidade. O projeto desta tela previu a possibilidade do campo em que o analista entra com o nome do componente possui o recurso de *auto complete* quando são recuperados ACLS já criadas previamente. Quando o analista clica em um dos componentes listados pelo *auto complete* o sistema mostra a relação de passos que constituem a ACL escolhida. Este recurso foi implementado nas telas de edição de procedimentos, rotina de verificação de pré-condição e de resultados esperados, o recurso de exibição do conteúdo da ACL foi implementado apenas nas telas de edição de ACL, podendo ser estendido para outras telas do sistema.



Figura K-4 - Tela de Edição de ACLS (Componentes).

Outras propostas de telas foram desenvolvidas como a tela de Cenários abaixo (Figura K-5) em que já aparece o relacionamento em árvore entre Casos de Uso, Cenários e Casos de Testes. Algumas telas propostas nesta etapa evoluíram até a fase de construção e testes, ficando disponível para uso em 12/12/2012. Sendo que em 02/01/2013 parte da equipe da organização foi treinada no uso da ferramenta para que fosse possível a execução do projeto piloto contemplando as mudanças no processo de testes e o uso do editor.

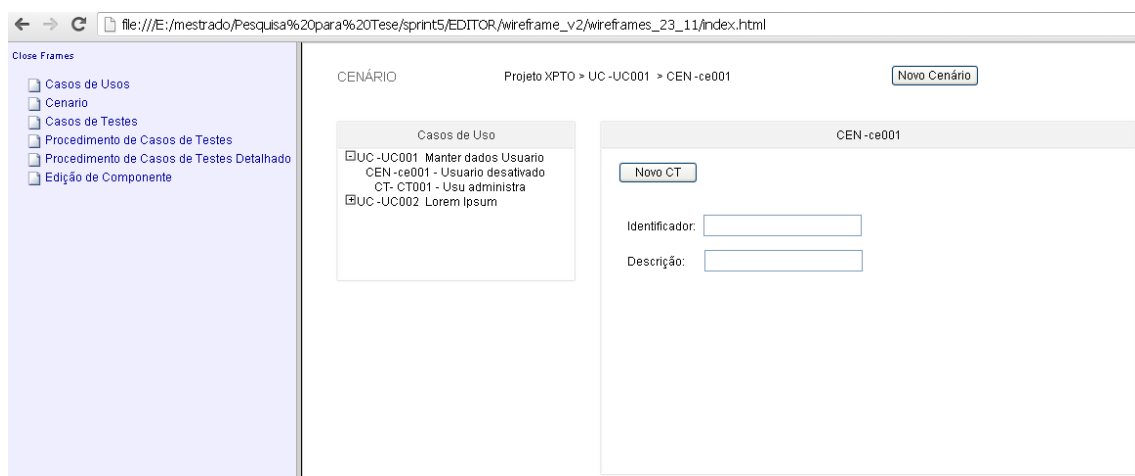


Figura K-5 - Edição de Cenários

K.2.4 Construção e Testes do Editor

Para a etapa de construção do Editor foi escolhido o SGBD Mysql e a linguagem Ruby on Rails 3.9.2. A escolha do SGBD foi devido ao largo uso deste gerenciador de banco de dados na indústria possuindo uma versão gratuita para download em site da internet. Já a escolha da linguagem Ruby foi devido às facilidades de geração de telas e de modelos a partir de uma sequencia de comandos pré-definidos.

Na etapa de testes a ferramenta foi instalada em um ambiente *Linux*, sendo utilizada a partir do módulo *passenger* do servidor de aplicação APACHE2. Os testes desenvolvidos para a ferramenta não chegaram a ser cadastrados no próprio Editor após o encerramento do desenvolvimento. Entretanto, pretende-se futuramente, incorporar ao pacote de instalação da ferramenta um projeto demo, contendo os Casos de Testes utilizados para verificar as funcionalidades da ferramenta.

Após a construção e os testes a atividade seguinte foi o planejamento do treinamento da equipe envolvida. Para isto todo o conjunto das telas utilizadas em cada uma das funcionalidades da ferramenta foi capturado e reproduzido em formato PPT para compor o material de treinamento. As principais telas do Editor estão dispostas no próximo item, algumas das telas exibidas foram extraídas diretamente do material de treinamento e outras foram recuperadas da execução do estudo de caso realizado na organização. O arquivo PPT foi incluído no pacote de instrumentação do estudo de caso.

K.2.4.1 Editor de Casos de Teste - Menus da Ferramenta

Após a entrada de login e senha para autenticação do usuário é exibida a tela inicial do sistema e, a lista de requisitos do release mais recente de um dos projetos no qual o usuário foi associado a um perfil de atuação. Sendo as opções de menu mostradas de acordo com este perfil. O menu de execuções por *default* só fica disponível para os testadores e gerentes de Testes. Os menus de desenvolvimento e testes são visíveis para todos os membros da equipe, sendo que apenas os usuários com perfil de analista de testes podem cadastrar e manter requisitos, cenários, casos de uso, casos de testes, procedimentos e variáveis.



Figura K-6 - Tela Principal do Editor de Casos de Testes (Perfil Gerente de testes)

Um determinado usuário cadastrado na ferramenta pode estar associado ao perfil de analista em um determinado release de um projeto e associado ao perfil de gerente de testes em outro release de um projeto. No exemplo acima o usuário autenticado no sistema possui o perfil de gerente de testes para a release 4 do projeto selecionado. Por isso os três menus são exibidos (Figura K-6).

No topo da tela ficam disponíveis duas *combo boxes* contendo as informações de projeto e releases cadastrados na ferramenta. De acordo com a seleção de projeto e release não apenas a lista de requisitos é atualizada como também os menus disponíveis.

Por exemplo, ao mudar o foco para outro release no qual está associado como analista de teste a exibição do Menu “Execuções” é inibida, refletindo as permissões de acesso deste perfil (Figura K-7).



Figura K-7 - Tela Principal do Editor de Casos de Testes (Perfil Analista de testes)

O Menu “Desenvolvimento” contém as opções para cadastramento dos projetos, releases, requisitos, regras de negócio, planos de testes e as funções de mapeamento e visualização da rastreabilidade dos elementos cadastrados (incluindo os artefatos de teste) e dos componentes (ACLS) desenvolvidos para reuso (Figura K-8).



Figura K-8 - Opções do Menu Desenvolvimento

O segundo Menu disponível é o Menu de Testes (Figura K-9).



Figura K-9 - Opções do Menu Testes

Este Menu contém as opções de edição e manutenção de Casos de Testes, Procedimentos de execução dos CTs, Procedimentos de verificação das pré-condições, Procedimentos de Verificação dos resultados esperados, Variáveis utilizadas nos procedimentos, componentes (ACLS) e Relatório contendo os Casos de Testes cadastrados com seus procedimentos, entradas, resultados esperados, e pré-condições (Figura K-9).

O Menu “Execuções” contém a função responsável pelo registro dos resultados de testes e está disponível para os usuários com perfil de gerente de teste e testador. Os demais podem consultar o resultado dos testes a partir da opção “Lista Projeto de Casos de Teste” do Menu “Testes”.

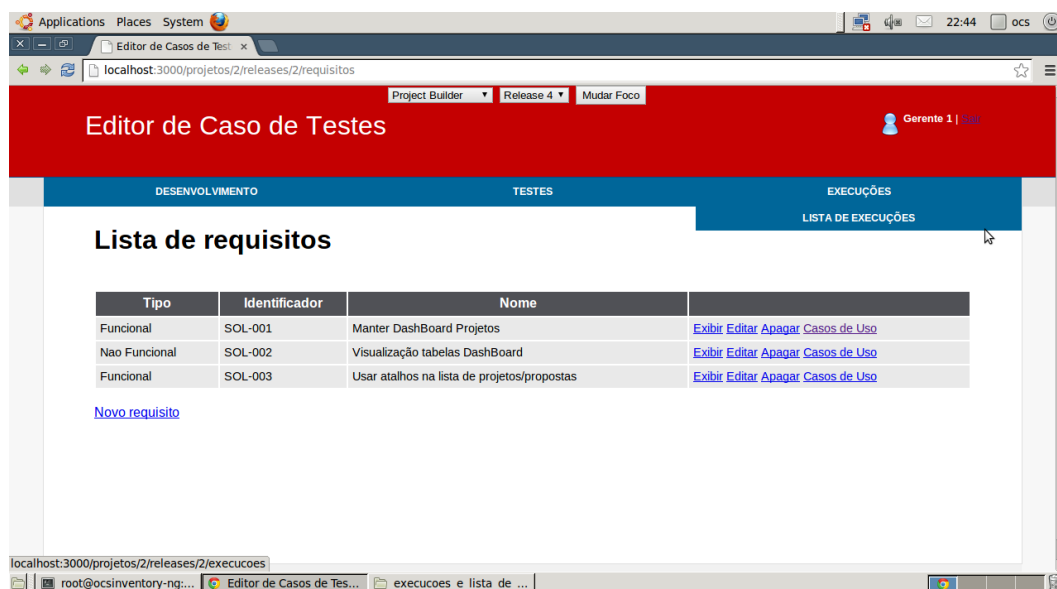


Figura K-10 - Opções do Menu “Execuções”

K.2.4.2 Projetos, Releases, Requisitos, Casos de Uso e Cenários.

As telas de manutenção dos elementos envolvidos no projeto de Testes seguem um padrão: para cada entidade do modelo (projeto, *release*, requisito, caso de uso, cenário, caso de teste, procedimento, resultado esperado, variável de entrada, pré-condição, componente) foi desenvolvido uma tela responsável pela exibição em formato de lista das entidades cadastradas no repositório da ferramenta e ao lado de cada linha as opções “Exibir”, “Editar” e “Apagar” para que o usuário possa manter o elemento exibido (Figura K-11).

Para a criação e edição de Casos de Testes o sistema oferece dois esquemas de navegação: (a) um esquema sequencial em que o usuário navega tela a tela através de *links* para realizar o cadastramento dos requisitos, regras de negócio, casos de uso, cenários, casos de teste e seus procedimentos (pré-condições, dados de entrada e resultados esperados) de forma *top-down*. (b) um esquema de acesso direto em situações em que os requisitos, regras de negócios e casos de Uso já foram criados, tornando possível através de uma lista encadeada em formato de uma árvore hierárquica criar ou editar um novo Caso de teste associando-o a um determinado cenário de uma forma mais direta, e também criar cenários associando-os a Casos de Uso.

Além disso, a ferramenta permite duas formas de cadastramento dos Casos de Testes e dos procedimentos: (a) o analista cadastra os atributos básicos do Caso de Teste (identificador, objetivo, descrição) e depois cria o procedimento, resultado esperado, pré-condição e as variáveis a serem utilizadas na seção de entradas do CT (*data pool*), para só então associar estes elementos ao Caso de Teste recém-criado (b) o analista cadastra primeiro os procedimentos, resultados esperados, dados de entrada (*data pool*) e pré-condição e na etapa de criação do Caso de Teste estabelece uma associação com os procedimentos, resultados esperados e pré-condições criados anteriormente.

K.2.4.3 Criando Projetos, Releases, Requisitos, UCs, Cenários e CTs

Para o cadastramento dos requisitos deverá ser necessário que o gerente de testes crie primeiro um projeto e um release. Para isto inicialmente deverá ser escolhida a opção “Projeto” do Menu Desenvolvimento, quando uma tela para exibição dos atributos de cada projeto é mostrada (Figura K-12).



Figura K-11 - Tela Lista de Projetos

Esta tela possui um link para criação de um novo Projeto (ao final da lista de projetos criados) e três links responsáveis respectivamente pela exibição dos atributos de cada projeto, edição e remoção.



Figura K-12 - Tela de exibição dos atributos de Projeto e Releases existentes

Ao ativar o *link* “Exibir” é exibido o nome do projeto, sua descrição, os releases previamente criados, sendo oferecida a opção de inclusão de um novo release (Figura K-12). Ainda na tela de lista de Projetos, quando o usuário ativa o *link* “Editar” uma tela para edição dos atributos é exibida (Figura K-13) permitindo a atualização dos campos nome e descrição.



Figura K-13 - Tela de edição de Projeto

Ao acionar o *link* “Novo Projeto” exibido ao final da tela de lista de Projetos (Figura K-10) o sistema formata uma tela para a especificação de um novo Projeto (Figura K-14).

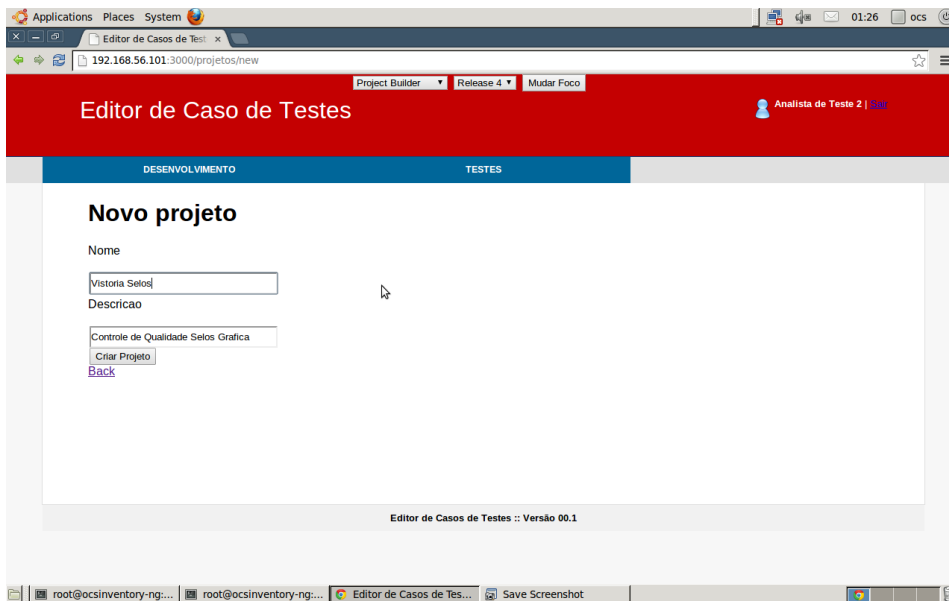


Figura K-14 - Tela de criação de Novo Projeto

As mesmas Telas de lista, edição e criação foram desenvolvidas para manutenção dos releases. Sendo que ao ser solicitada a opção “Release” do Menu de Desenvolvimento só são mostrados os releases do projeto selecionado no combo box de Projetos disposto no topo da tela. Neste caso são mostrados apenas os releases do projeto selecionado “Vistoria Selos” (Figura K-15).

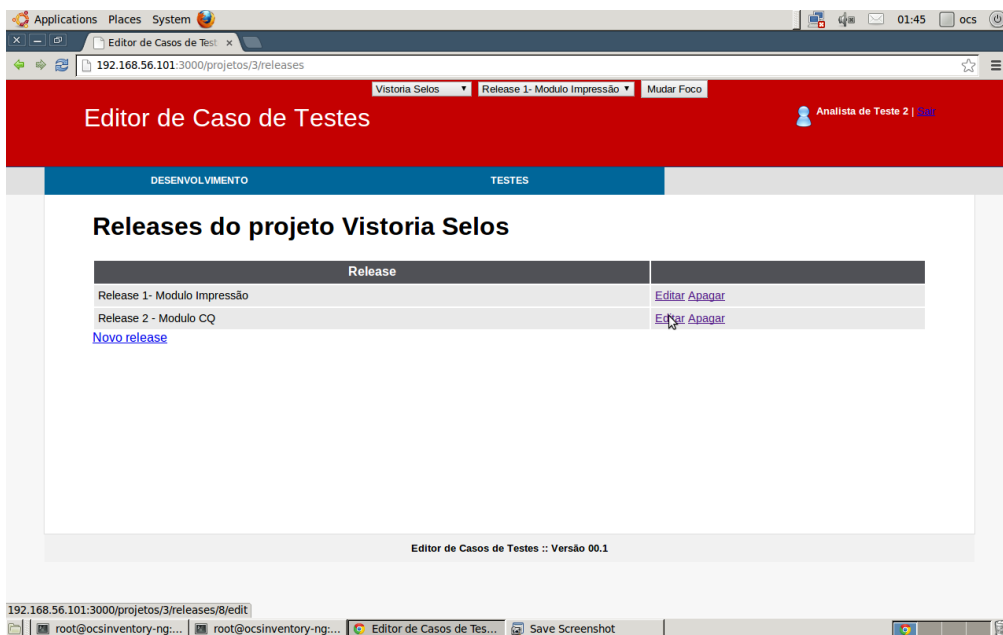


Figura K-15 - Lista de Releases do projeto selecionado

Para o cadastramento de um requisito o usuário deverá selecionar o projeto e release para o qual o requisito será associado, e quando a tela de lista de requisitos for exibida clicar no *link* “Novo requisito”. Conforme mostrado na Figura K-16, no “Release 1 – Modulo de Impressão” do projeto “Vistoria Selos” selecionado já existiam 2 requisitos funcionais especificados.



Figura K-16 - Listas de Requisitos de um Release de um Projeto

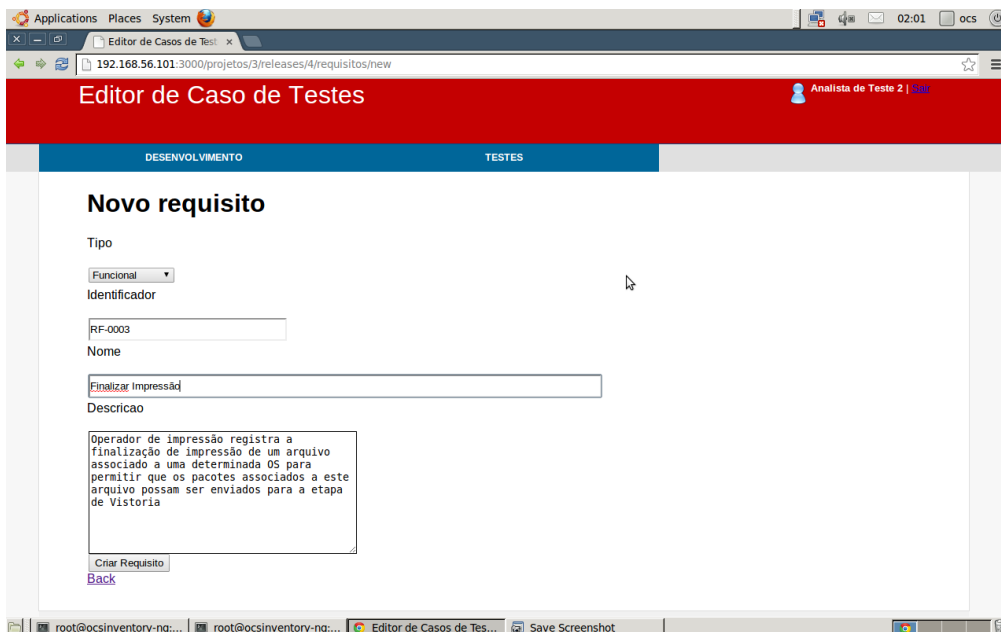


Figura K-17 - Novo Requisito

Uma tela é exibida para que o usuário entre com informações sobre o novo requisito, quando deverá ser informado se o requisito é funcional ou não funcional, um

código que facilite sua identificação, um nome e opcionalmente uma descrição mais detalhada (Figura K-17).

A tela de Lista de requisitos do projeto em foco é mostrada novamente com o novo requisito incluído na lista. Caso o usuário selecione um novo Projeto e pressione o botão “Mudar Foco” - conforme mostrado na figura K-18 – uma nova lista de requisitos é atualizada na tela da ferramenta (Figura K-19).



Figura K-18 - Trocando de Foco de Projeto

A Figura K-19 exibe a lista de requisitos do novo projeto selecionado.

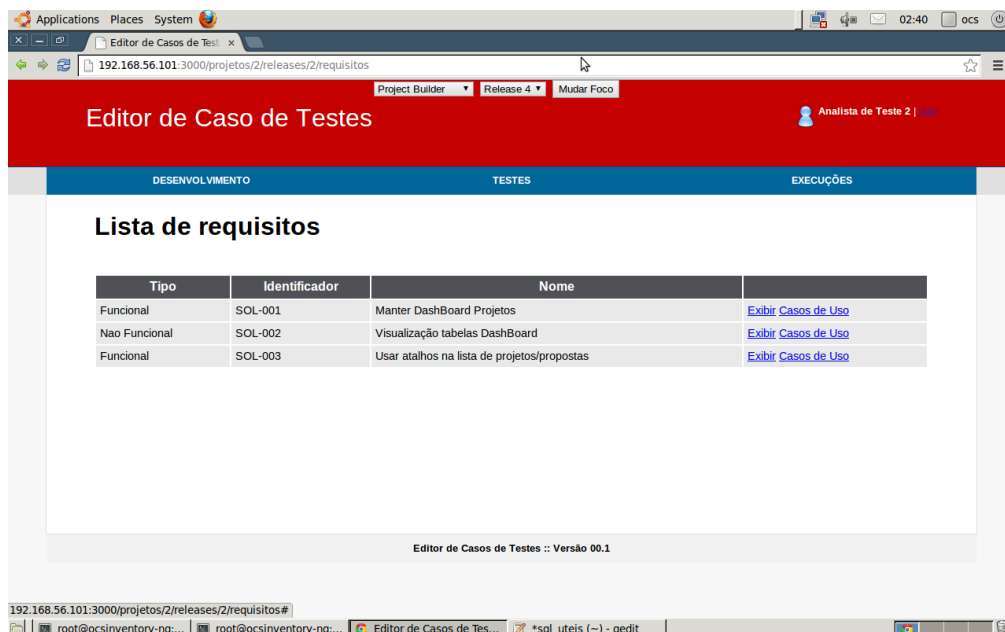


Figura K-19 - Cadastrando um Caso de Uso a partir da Tela de Lista de Requisitos

Para cadastrar um Caso de Uso associado a um requisito deverá ser executado um procedimento similar, sendo que deverá ser acionado o *link* Casos de Uso disposto ao lado de cada requisito na tela de lista de requisitos.

Como já existia um Caso de Uso (UC-133) associado ao requisito SOL-001 a tela exibida como resposta do acionamento do link mostra este caso de uso associado ao requisito SOL-001 (Figura K-20).



Figura K-20 - Solicitando tela de cadastramento de Caso de Uso

Da mesma forma que na etapa de criação de um novo requisito uma tela é exibida para que o usuário entre com as informações relativas ao caso de uso que será cadastrado (Figura K-21).

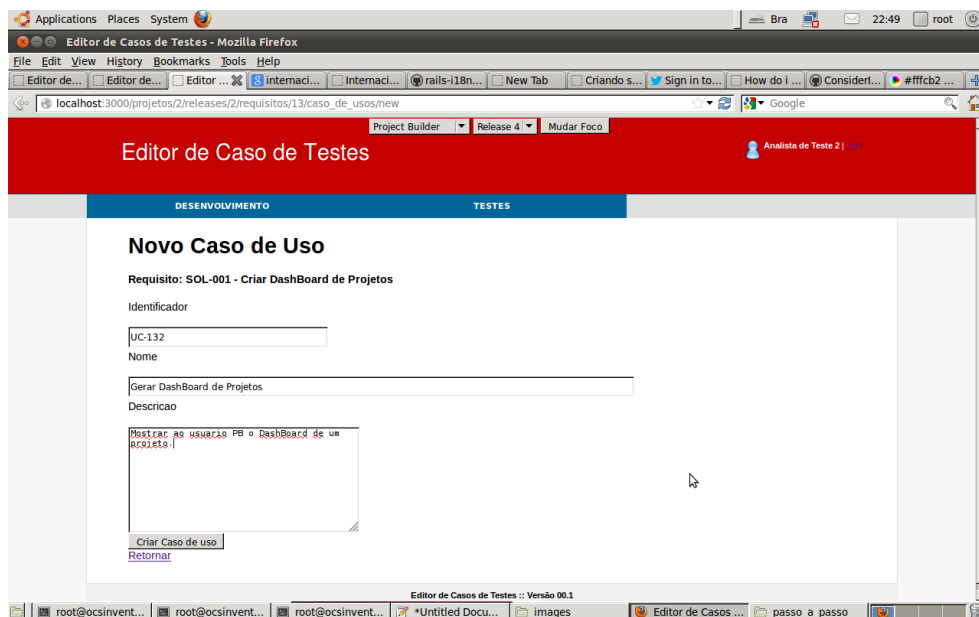


Figura K-21 - Tela criação Novo Caso de Uso

Após o usuário comandar a criação a partir da submissão da tela de cadastramento de Caso de Uso o sistema retorna para a tela de lista de Casos de Uso (já atualizada com o novo Caso de Uso recém-criado). Da mesma forma que na criação do Caso de Uso um cenário poderá ser criado associado a um dos UCs (no caso o UC-132) bastando para isso que o usuário clique no *link* “Cenários”.



Figura K-22 - Solicitando Tela Criação de Novo Cenário

Após o novo cenário ter sido criado (em nosso exemplo CEN-001), a tela de Lista de cenários é exibida e através do *link* Caso de Testes o analista poderá solicitar a criação de um Caso de Teste (Figura K-23).

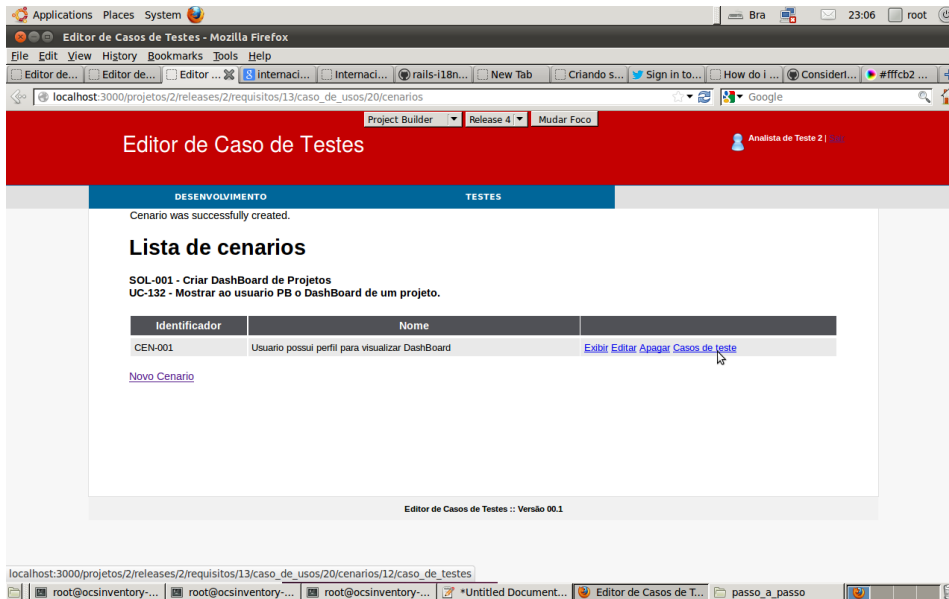


Figura K-23 - Solicitando Tela Criação de Novo Caso de Teste

Como o cenário foi recém-criado, ainda não existe nenhum caso de teste associado. E neste caso a tela de lista de Casos de Teste ainda não mostra nenhum CT associado (Figura K-24).



Figura K-24 - Tela de Lista de Casos de Teste

O usuário clica no *link* Novo Caso de Teste e o sistema exibe a tela para que seja incluído o novo Caso de Teste. Todo o Caso de Teste deve ser associado a um determinado plano a ser criado através da opção Planos do Menu de Desenvolvimento. Neste exemplo estará sendo utilizado um Plano de Testes de Sistemas denominado PL-SIS-001 criado previamente.

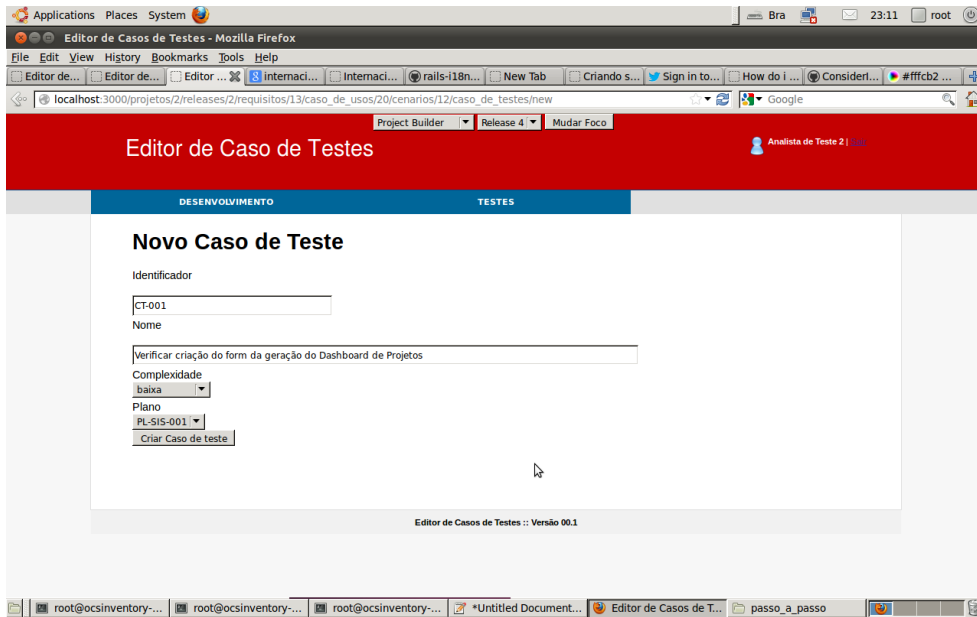


Figura K-25 - Criação de Novo Caso de Teste

Ao comandar a criação do Caso de Teste uma nova Tela é exibida (Figura K-26) para que o analista entre com os elementos que irão compor o Caso de Teste: os nomes de procedimentos, pré-condições, resultados esperados e associar às variáveis criadas previamente os valores de entrada que irão compor o Caso de Teste. Caso o usuário ainda não tenha criado estes elementos, deverá acessar o menu Testes para solicitar a criação.

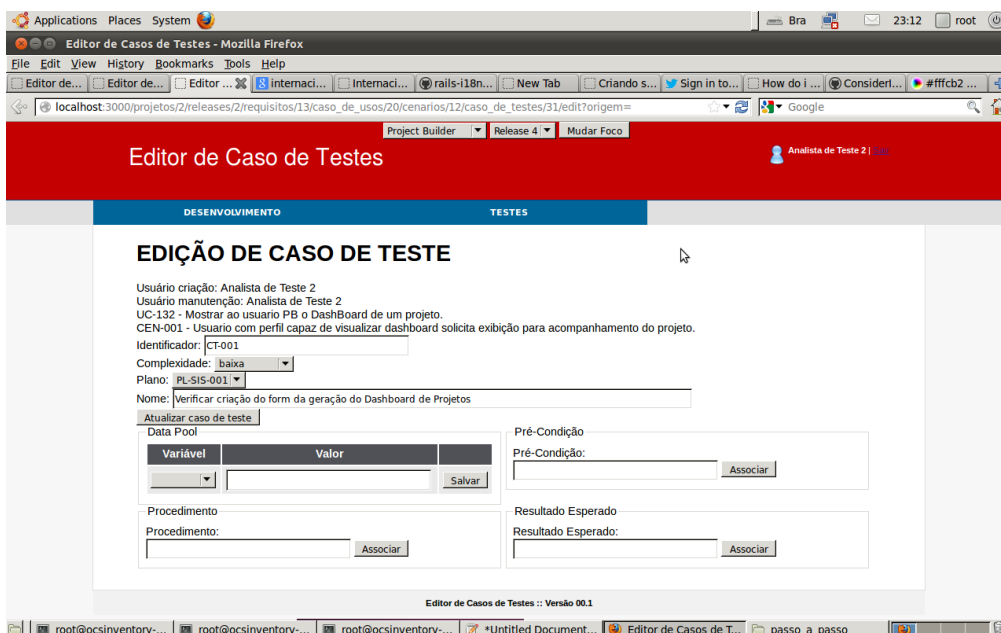


Figura K-26 - Tela de Edição de Casos de Testes

No tópico K.2.4.5 é demonstrado passo-a-passo como executar o cadastramento de procedimentos, resultados esperados e pré-condições.

K.2.4.4 Usando um mapa para visualizar Casos de Teste

Outra forma de realizar o cadastramento de CTs é solicitar a opção “Mapa de Casos de Testes” e acessar diretamente o Cenário ou Caso de Teste a ser editado.



Figura K-27 - Tela Mapa de Casos de Testes

Esta forma pode ser bastante útil quando os requisitos e casos de Uso já foram cadastrados e é preciso apenas cadastrar o cenário, ou apenas criar ou editar um caso de Teste. Na figura K-27 é exibida a tela enviada quando é escolhida esta opção.

Ao clicar no nome do CT (CT-001 Verificar criação do *Form* da geração de *Dashboard* de Projetos) o sistema exibe no frame à direita os atributos do caso de teste selecionado para edição.



Figura K-28 - Tela de Edição de Caso de Teste e Mapa

A partir desta tela o caso de teste pode ser editado da mesma forma como se fosse acessado através da tela de lista de Casos de Testes.

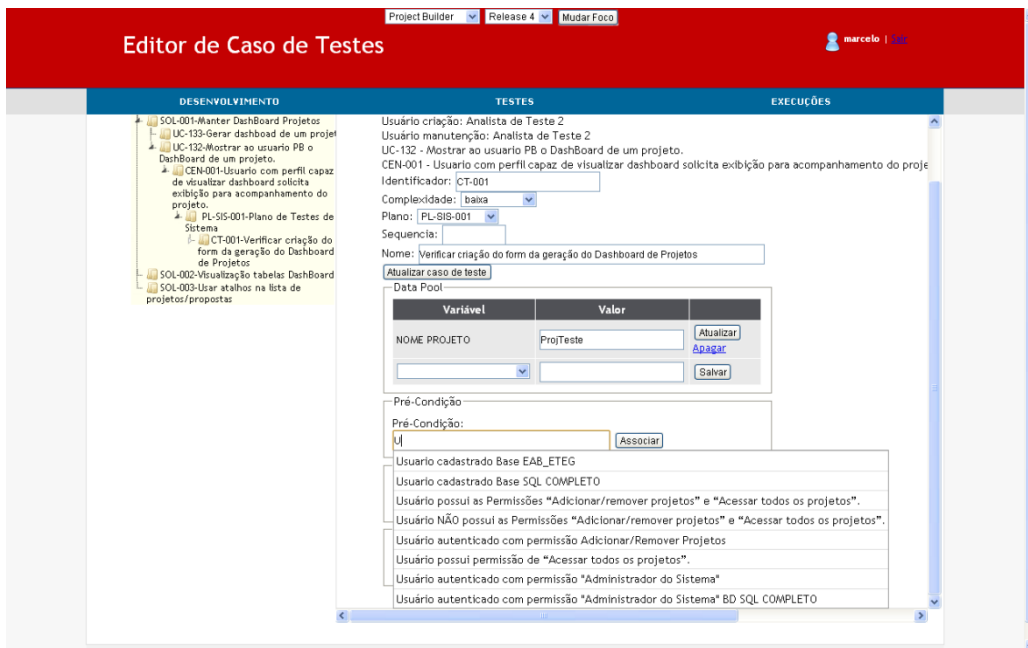


Figura K-29 - Associando uma pré-condição criada previamente a um Caso de Teste

No CT utilizado como exemplo apenas a variável a ser utilizada na entrada de dados foi especificada, faltando associar uma pré-condição, um procedimento para execução do CT e uma rotina de verificação de resultado esperado.

Na figura K-29 é exibido o recurso de *auto complete* utilizado na seleção de uma rotina de pré-condição definida previamente a partir da lista de pré-condições existentes para o release em foco.

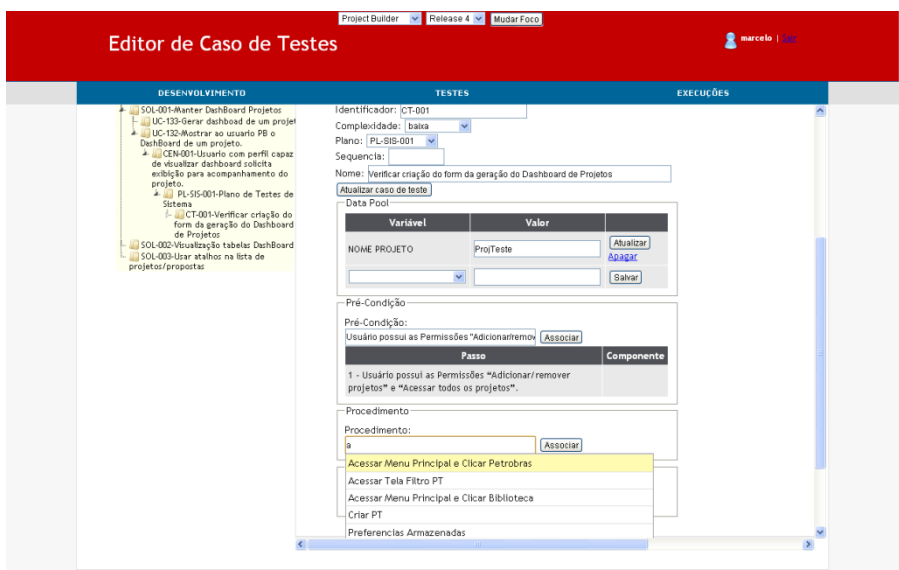


Figura K-30 - Associando um procedimento a um Caso de Teste

Após o usuário definir a pré-condição é retornada a tela de edição de CT para que sejam associados os procedimentos e resultados esperados criados previamente.

O Caso de Teste pode ser acessado novamente a partir do caminho Requisitos -> Casos de Uso-> Cenários -> Casos de Testes *link* “Editar” da tela Lista de Casos de Testes (Figura K-31).



Figura K-31 - Acessando CT a partir de Tela Lista de Casos de Testes

.Na figura K-32 é mostrado o mesmo Caso de Teste que foi editado originariamente a partir do mapa de Casos de Testes. Nesta forma de apresentação existe mais espaço disponível para visualizar os elementos que compõem o CT.

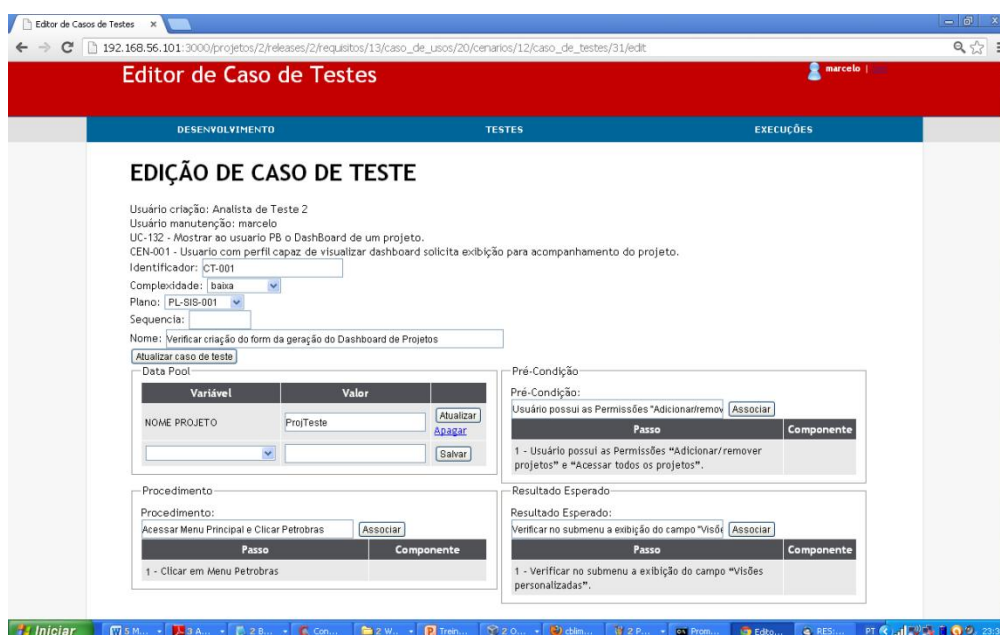


Figura K-32 - Tela de Edição de Caso de Teste

Após associação de uma pré-condição, procedimento de execução, resultados esperados e variáveis que irão ser utilizadas na entrada de dados ao Caso de Teste recém-criado, todos estes elementos podem ser visualizados a partir da opção Mapa de Casos de Testes do Menu Desenvolvimento.



Figura K-33 - Rastreabilidade de Requisitos até elementos de Caso de Teste

Usando esta estrutura hierárquica é possível solicitar a criação de um novo Caso de Teste acessando qualquer um dos cenários mostrados no mapa de Caso de Testes (Figura K-33) e escolhendo a opção “Criar novo caso de teste”.

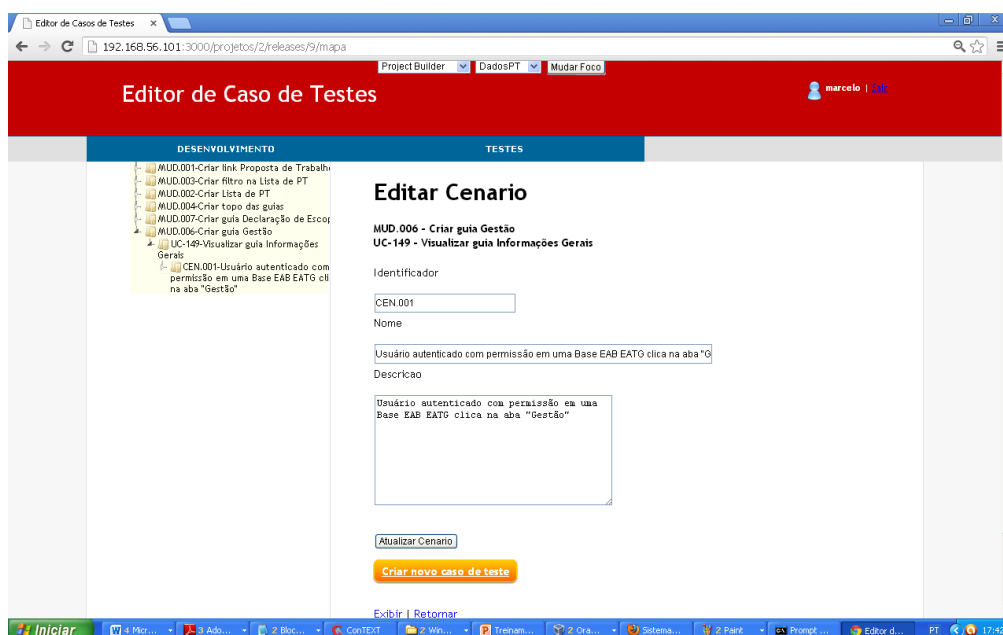


Figura K-34 - Criação de Caso de Teste a partir de Cenário exibido no Mapa de Artefatos de teste

Além da opção de acessar o caminho de relacionamento entre requisitos, casos de uso, cenários, casos de testes através da opção Mapa Procedimentos é possível caminhar no sentido contrário procedimento -> Caso de Testes.



Figura K-35 - Rastreamento de Procedimentos a Casos de Testes

A figura K-35 exibe o relacionamento entre um procedimento e um conjunto de Casos de Teste que o reutiliza.

K.2.4.5 Criando uma ACL e uma rotina de Pré-Condição

Conforme definido na etapa de análise e elicitação de requisitos, uma ACL é um componente constituído de um conjunto de ações previamente cadastradas que poderão ser associadas a um passo de um destes elementos.

Esta lista de ações (ACL) poderá ser criada através do Editor para ser reutilizada por elementos de cada Caso de Teste (procedimentos, pré-condições e resultados esperados).

O analista de teste ao descrever cada passo de cada elemento do CT (pré-condição, procedimento p/ execução, verificação de resultado esperado) tem a opção de associa-lo a um componente (ACL) previamente cadastrado.

No exemplo abaixo são exibidas as telas utilizadas na criação de um determinado componente e sua utilização em uma rotina de pré-condição desenvolvida durante a execução do estudo de Caso realizado na organização.

Inicialmente o usuário no Menu Testes solicita a opção Componentes (ACLS) e o sistema exibe a Lista de ACLS disponíveis.

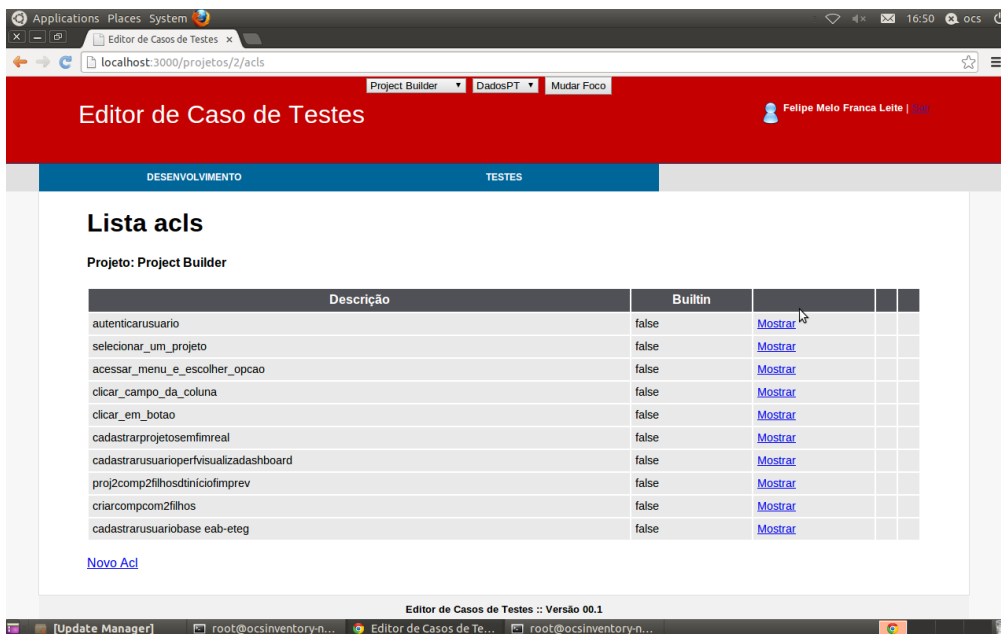


Figura K-36 - Lista de Componentes disponíveis

O usuário deverá acessar o link “Novo ACL” e uma tela será exibida para que o usuário entre com a relação de ações que irão fazer parte deste componente.



Figura K-37 - - Criação de uma ACL

Os componentes cadastrados podem ser utilizados por um Caso de Teste em qualquer uma das rotinas de verificação de resultado esperado, pré-condição e procedimento de execução de Caso de Teste.

No caso do componente criado “Cadastrarusuariobaseeab_eteg” pelo fato de suas ações estarem associadas a procedimentos de preparação de dados para a realização de um CT ele estará sendo associado a uma pré-condição.

Para utilizar este componente em uma rotina de pré-condição inicialmente deverá ser cadastrada uma rotina deste tipo. Para acessar a tela de criação de Pré-Condição o usuário escolhe a opção “Pré-Condições” do Menu Testes e acessa o link “Nova Pré-Condição” quando será apresentada uma tela para entrada de dados (Figura K-38).

The screenshot shows a web browser window displaying the 'Editor de Caso de Testes' application. The page title is 'Nova Pré-Condição'. The form contains the following fields and content:

- Identificador:** A text input field containing 'PRE-007'.
- Nome:** A text input field containing 'Usuário possui permissão de "Acessar todos os projetos"'. Below the field is the label 'Descricao'.
- Descricao:** A text area containing the text: 'Usuário autenticado no sistema deverá possuir permissão de acesso a todos os projetos cadastrados na ferramenta de GP.'.

At the bottom of the form area, there are two buttons: 'Criar Pré condicao' and 'Retornar'. The footer of the application indicates 'Editor de Casos de Testes :: Versão 00.1'.

Figura K-38 - Tela de Criação de Nova Pré-Condição

Após o envio da tela de criação da pré-condição o sistema exibe uma tela para edição dos passos que irão constituir o procedimento de pré-condição (Figura K-39). O formato desta tela é o mesmo para os procedimentos de execução do CT, as rotinas de verificação de resultados esperados e as rotinas de preparação do ambiente para atendimento a pré-condições. No topo da tela é disposto um conjunto de campos com os atributos básicos do procedimento como identificador, Nome e descrição.

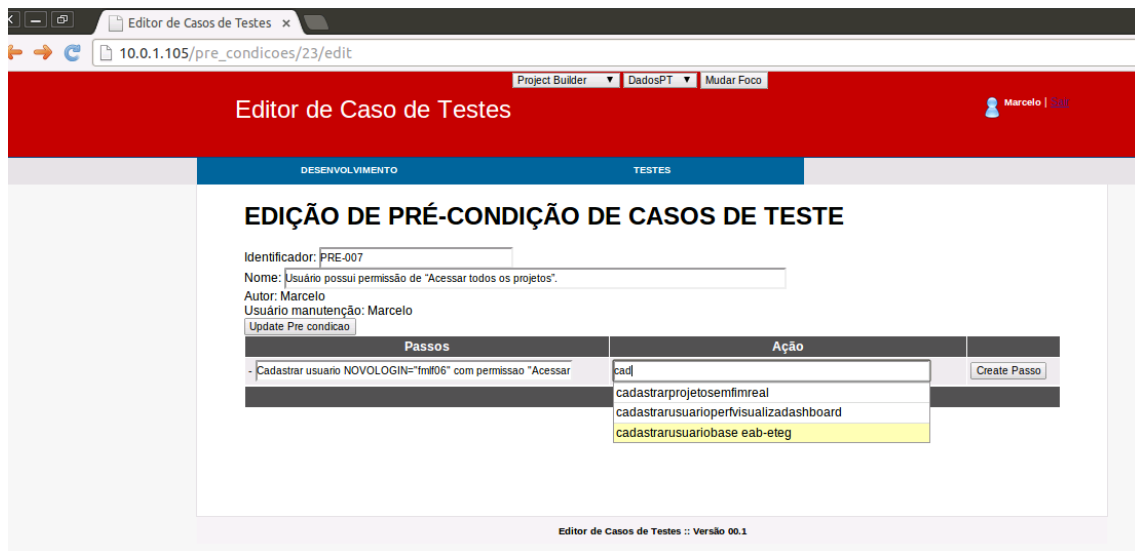


Figura K-39 - Associando um componente criado previamente a um passo de pré-condição

E logo abaixo três colunas para preenchimento pelo usuário: (a) em passos o analista descreve em texto livre o que consiste o passo necessário para preparação e atendimento da pré-condição, (b) em ação ele pode opcionalmente associar a um componente cadastrado previamente, sendo implementado nesta coluna o recurso de auto complete (c) uma terceira coluna para que o usuário comande a criação, atualização, exclusão de um passo ou inserção de um passo acima do passo corrente.



Figura K-40 - Exibindo conteúdo de Componente a ser associado a pré-condição

A qualquer momento é possível visualizar o conteúdo de um componente bastando para isso clicar em uma das ACLs listadas. Sendo possível ainda, listar o conteúdo de cada subcomponente clicando no botão disposto ao lado (Figura K-40).

K.2.4.6 Criando Procedimentos, Resultados Esperados e Variáveis.

Da mesma forma que na criação de pré-condição, para acessar a tela de criação de Procedimento, o usuário escolhe a opção “Procedimentos” do Menu Testes e acessa o link “Novo Procedimento” quando será apresentada uma tela para entrada de dados. Após a entrada do id, nome e descrição, o sistema retorna uma tela para que sejam informados os passos e opcionalmente componentes associados (Figura K-41).

Editor de Caso de Testes

DESENVOLVIMENTO TESTES

EDIÇÃO DE PROCEDIMENTO DE CASOS DE TESTE

Identificador: PR-004
Nome: Acessar Tela Filtro PT
Autor: marcelo
Usuário manutenção: marcelo
Update Procedimento

Passos	Ação
1 - No menu principal clicar na opção "Petrobras"	Update Passo Apagar Incluir acima
2 - No submenu clicar na opção "Lista de PT";	Update Passo Apagar Incluir acima
3 - Na página de "Lista Pts" clicar na aba "Filtro";	Update Passo Apagar Incluir acima
-	Create Passo

Parâmetros

Editor de Casos de Testes :: Versão 00.1

Figura K-41 - Cadastrando passos de um procedimento

Nesta tela é mostrado um procedimento em que o analista cadastrou cada um dos passos, porém optou por não associa-los a nenhum componente.

Na figura K-42 é mostrado um procedimento de verificação de resultado esperado. O analista optou também por não utilizar componente previamente cadastrado.

Editor de Caso de Testes

DESENVOLVIMENTO TESTES

EDIÇÃO DE RESULTADO ESPERADO DE CASOS DE TESTE

Identificador: RES-007
Nome: Verificar se as PTs retornadas possuem o status de programa de trabalho
Imagem: [Choose File](#) No file chosen
Autor: marcelo
Usuário manutenção: marcelo
Update Resultado esperado

Passos	Ação
1 - Verificar se as PTs retornadas possuem status STATUSSEL.	Update Passo Apagar Incluir acima
-	Create Passo

Parâmetros

Editor de Casos de Testes :: Versão 00.1

Figura K-42 - Cadastrando passos de uma Rotina de Verificação de resultados Esperados

A ferramenta permite também associar a resultados esperados uma imagem a ser confrontada com o resultado obtido na execução do Caso de Teste. Para isto o analista deverá escolher uma imagem armazenada previamente em algum diretório da maquina local e estabelecer a associação (Figuras K-43 e K-44).

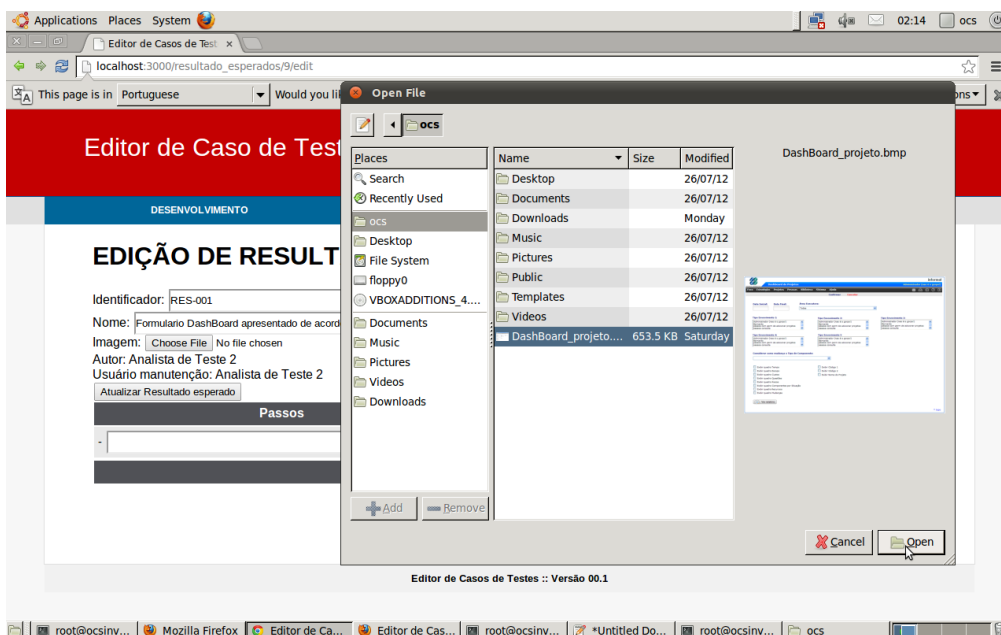


Figura K-43 - Associando uma imagem a um resultado esperado

Após escolher o arquivo contendo a imagem a ferramenta passa a mostrar conjuntamente com os passos de verificação do resultado esperado a imagem associada (Figura K-44).

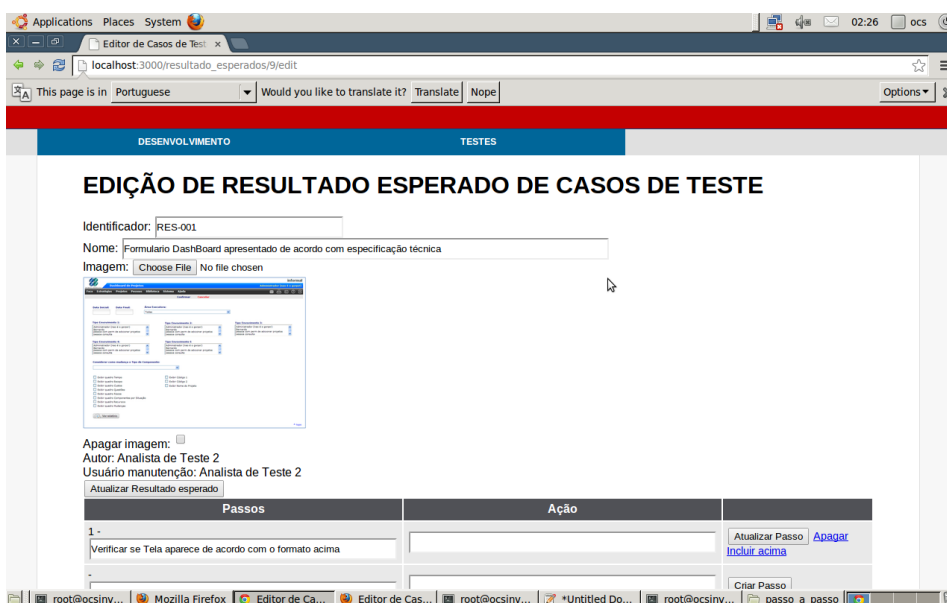


Figura K-44 - Exibindo Imagem associada a Resultado Esperado

Complementando a criação dos elementos a serem associados aos Casos de Testes a ferramenta prove uma funcionalidade para cadastramento das variáveis referenciadas nas entradas de cada CT e nos procedimentos. Para acessar esta função o analista deverá escolher a opção Variáveis no Menu de Testes e o sistema retornará uma tela contendo a lista de variáveis criadas para todo o projeto em foco.

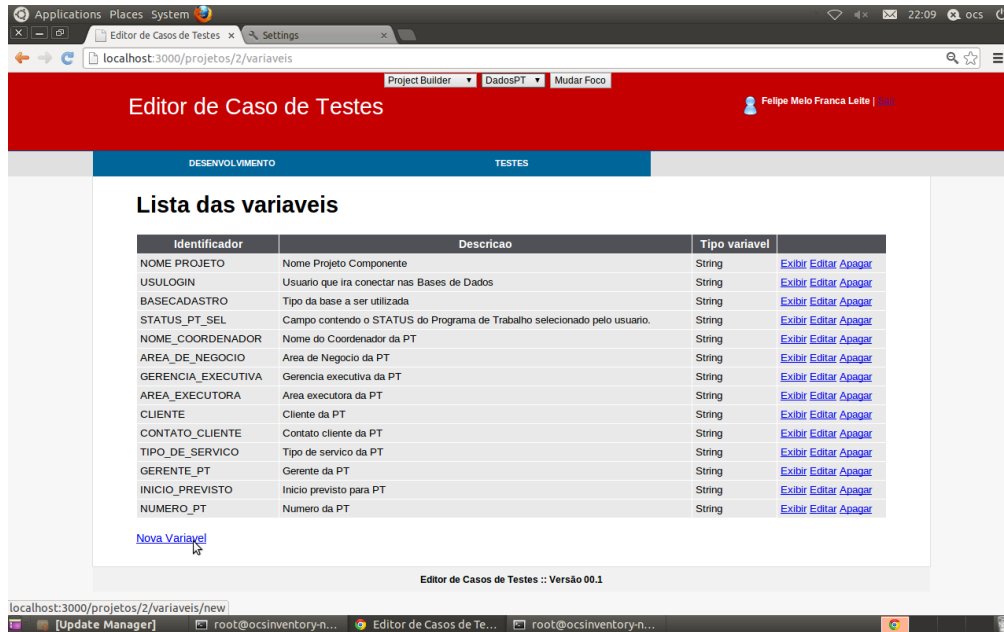


Figura K-45 - Lista de variáveis

Como resposta a solicitação de uma nova variável o sistema envia uma tela para que o analista possa especificar as características desta nova variável.

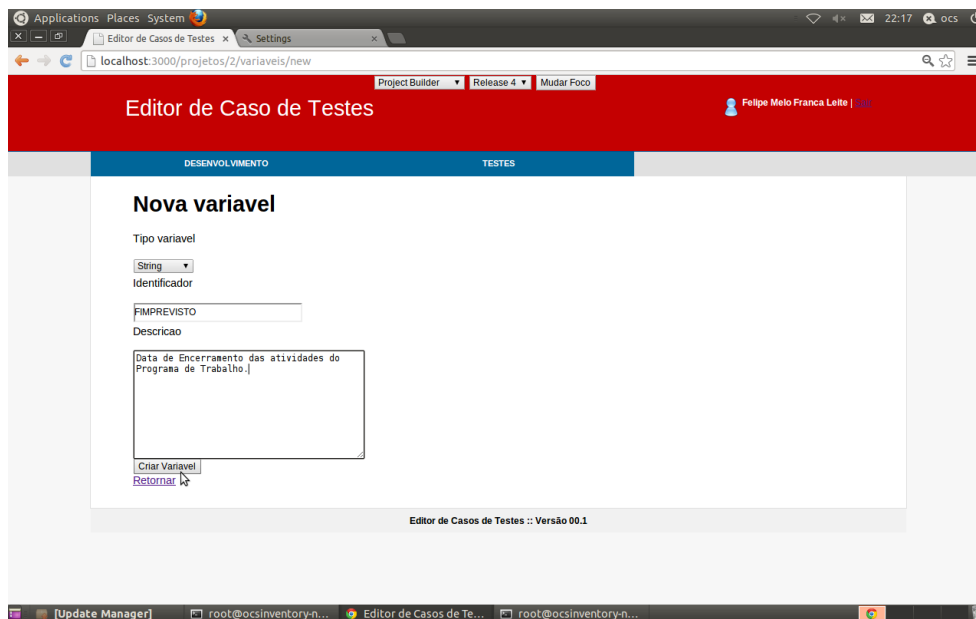


Figura K-46 - Criação de Nova Variável

K.2.4.7 Gerando o Plano de Testes e Registrando os Incidentes de Testes

De forma a facilitar a execução dos Casos de Teste o Editor permite a geração dos casos de testes associados a um determinado plano em formato PDF. Para isto o usuário deverá escolher a opção Lista de Casos de testes no menu Testes, quando será exibida uma tela em que estarão dispostos em uma combo box a lista dos planos definidos no âmbito do projeto como filtro (Figura K-47).



Figura K-47 - Tela de Lista de Projetos de Casos de Teste

Após ser escolhido o plano pelo usuário o sistema envia uma tela contendo a listagem de todos os Casos de teste associados ao Plano de teste selecionado, permitindo gerar o plano em formato PDF.

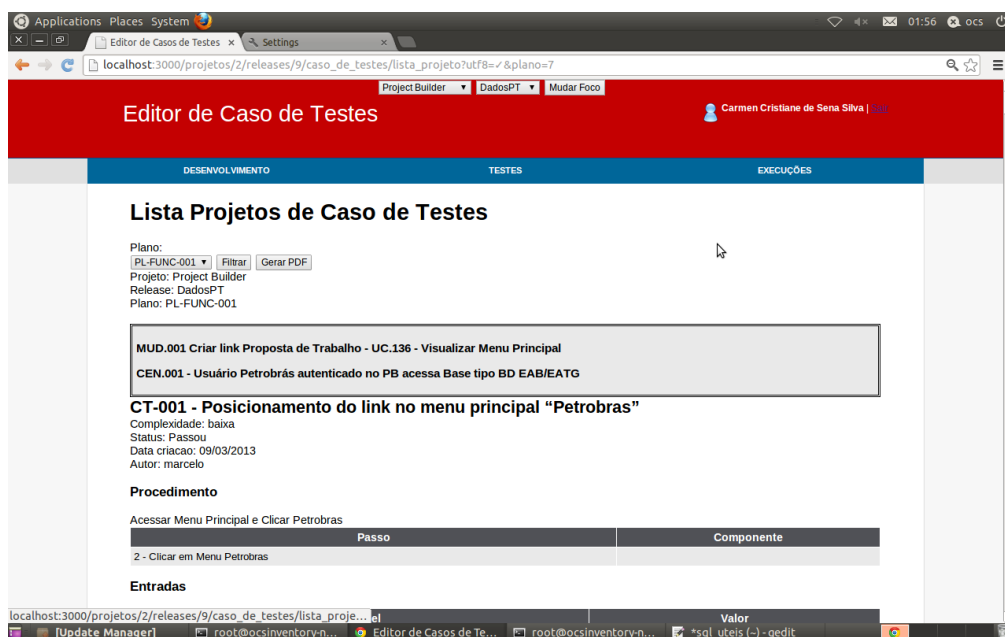


Figura K-48 - Lista de Projetos de Casos de Testes

Com o plano do projeto definido, o testador pode iniciar a execução dos testes e registrar cada execução através da opção “Casos de testes” do Menu Testes - quando será enviada uma tela contendo todos os Casos de Testes definidos para o Plano de teste utilizado como filtro (Figura K-49).

Identificador	Nome	Plano	Complexidade	Status	Data criação	Autor	
CT-001	Posicionamento do link no menu principal "Petrobras"	PL-FUNC-001	baixa	Passou	09/03/2013	marcelo	Alterar status Exibir
CT-003	Exibição das opções do item de menu Petrobras(Regra de Negócio RN1)	PL-FUNC-001	baixa	Passou	09/03/2013	Felipe Melo Franca Leite	Alterar status Exibir
CT-004	Exibição da opção "Relatórios" no item de menu "Petrobras"	PL-FUNC-001	baixa	Passou	09/03/2013	Felipe Melo Franca Leite	Alterar status Exibir
CT-005	Exibição da opção "Nova PT" no item de menu "Petrobras"	PL-FUNC-001	baixa	Falhou	09/03/2013	Felipe Melo Franca Leite	Alterar status Exibir
CT-006	Ausência Opção "Lista de PT" no item de menu "Petrobras"	PL-FUNC-001	baixa		11/03/2013	Felipe Melo Franca Leite	Alterar status Exibir
CT-007	Exibição das opções "Lista de PT", "Nova PT" e "Relatório" no item de menu "Petrobras"	PL-FUNC-001	baixa		11/03/2013	Felipe Melo Franca Leite	Alterar status Exibir
CT-008	Exibição da opção "Gestão LD" no item de menu de "Petrobras"	PL-FUNC-001	baixa	Passou	11/03/2013	Felipe Melo Franca Leite	Alterar status Exibir

Figura K-49 - Lista de Casos de testes

O testador pode acionar o link “Alterar Status” para registrar uma execução de teste e associar um status para o CT.

Caso de Teste: CT-007

Status:

Comentário:
Opção "Relatorio" não foi mostrada apesar de usuario possuir permissão "Acessar todos os projetos"

Data execução: 12 Maio 2013

Editor de Casos de Testes :: Versão 00.1

Figura K-50 - Registrando execução de Caso de Teste

Finalizando as etapas de projeto, construção e execução dos Testes, o registro das execuções pode ser visualizado e gerado em formato PDF a partir da opção “Lista de Execuções” do menu Execuções (Figura K-51).



Figura K-51 - Lista de Execuções por Caso de Teste

K.3 Conclusão

Neste apêndice foi apresentado um Editor de Casos de Testes cuja implementação visa facilitar a criação e reutilização de blocos de procedimentos, resultados esperados e pré-condições desenvolvidos durante a evolução dos releases de um determinado produto, com o intuito de mitigar eventual retrabalho na elaboração dos Casos de Testes Funcionais.

O objetivo é oferecer um mecanismo que facilite a busca e visualização dos blocos e Casos de Teste, de forma que os participantes de uma equipe de Teste possam compartilhar CTs desenvolvidos no decorrer da evolução de um determinado produto, que permita o analista de teste descrever o CT evitando um projeto com complexidade desnecessária e código extra, um princípio básico da prática Design Simple.

Durante a etapa de elicitação dos requisitos, além dos subsídios obtidos na realização das entrevistas e nos resultados obtidos na revisão sistemática em que foram recuperados artigos na academia sobre dificuldades encontradas na manutenção de Procedimentos e Casos de Testes foram também avaliados alguns dos recursos existentes em algumas ferramentas de mercado e como também na utilizada pela organização em suas atividades de projeto, construção e execução de Caso de Testes.

Algumas das características existentes nesta ferramenta foram incluídas no projeto do Editor proposto, desde que se mostrou necessário fornecer suporte para a inserção de práticas ágeis, como também para outras atividades que já eram rotineiramente desenvolvidas pela organização como, por exemplo, das funções relacionadas a registro de incidentes de testes e geração do plano de testes contendo a relação de Casos de Testes e procedimentos.

APENDICE L - Relação de Melhorias sugeridas para a ferramenta de Edição

Neste apêndice é apresentada uma lista de novas funcionalidades a serem incorporadas na To-do List da Ferramenta de Edição de Casos de Testes.

L1 – Lista

- 1- A ferramenta deverá permitir associar Casos de Testes já existentes a novos Planos para que seja possível re-executar Casos de Testes quando for necessário verificar a execução das funcionalidades em outros ambientes (por exemplo em outro SGBD, sistema operacional). Uma execução de um Caso de Teste é atributo de um determinado Plano, podendo ser possível associar um determinado caso de teste a outro Plano para que sua execução possa ser realizada a partir de outro contexto.
- 2- Deverá ser possível copiar Casos de Teste em lote para novos planos de novos releases.
- 3- A ferramenta não deverá permitir editar casos de testes que já tenham sido executados. Mas deverá permitir gerar novas versões após a realização de execuções.
- 4- A ferramenta deverá permitir o gerente de teste indicar quais testadores serão responsáveis por quais Casos de Testes.
- 5- A ferramenta deverá permitir visualizar o conteúdo das ACLs junto com os procedimentos que a utilizam. Atualmente o usuário precisa acessar a ACL para visualizar seu conteúdo.
- 6- A ferramenta deverá permitir ao usuário salvar o conteúdo de uma ACL com outro nome para permitir reaproveitamento de blocos de ações.
- 7- Para facilitar o registro de execuções a tela de registro de execução dos testes deverá mostrar os passos do procedimento dos Casos de Teste. Atualmente os passos são mostrados em uma tela à parte.
- 8- Deverão ser implementados filtros que permitam ao analista de teste listar Casos de Testes por status (executado com sucesso, executado com falha, bloqueado).
- 9- Ao associar um componente a um determinado passo de procedimento, pré-condição ou resultado esperado deverá ser possível especificar os valores para os parâmetros da execução do Caso de Teste.
- 10- Não foi prevista na etapa de criação dos Casos de Testes a possibilidade de definir a ordem de execução para apoiar as etapas de documentação e execução dos Casos de Testes.
- 11- Deverá ser possível associar mais de uma pré-condição a um mesmo Caso de Teste.

12- O módulo de relatório deverá ser capaz de gerar o status atual do progresso de testes.