


UMA INFRA-ESTRUTURA COMPUTACIONAL PARA APOIAR O PLANEJAMENTO E
CONTROLE DE TESTES DE SOFTWARE

Arilo Claudio Dias Neto

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA 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.

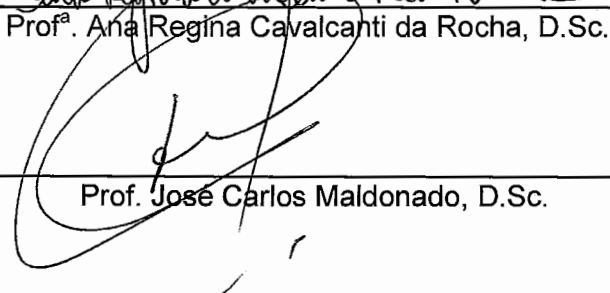
Aprovada por:



Prof. Guilherme Horta Travassos, D.Sc.



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



Prof. José Carlos Maldonado, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2006

DIAS NETO, ARILO CLAUDIO

Uma Infra-Estrutura Computacional para
Apoiar o Planejamento e Controle de Teste de
Software [Rio de Janeiro] 2006

IX, 154 p. 29,7 cm (COPPE/UFRJ, M.Sc.,
Engenharia de Sistemas e Computação, 2006)

Dissertação - Universidade Federal do Rio de
Janeiro, COPPE

1. Teste de Software
 2. Planejamento e Controle de Teste de Software
 3. Infra-estrutura Computacional
- I. COPPE/UFRJ II. Título (série)

Aos meus Pais, meus exemplos de vida e de onde tiro minha inspiração.
À minha Irmã, Danielly, à minha Sobrinha Lindsen, apesar da distância, e à Dessana.

Agradecimentos

Primeiramente a Deus, por estar comigo em todos os momentos e por ter me dado saúde e o dom da sabedoria. Nada seria possível sem Ele.

À minha Mãe, Roseneide, pelo amor, compreensão e apoio irrestrito em todos os momentos dessa longa jornada.

Ao meu Pai, José Ribamar, pelo carinho, força e por todo exemplo de vida que sempre me deu.

À minha Irmã, Danielly, e à minha sobrinha, Lindsen, pela força e inspiração. Não há nada mais gratificante que ver o sorriso de uma criança.

À minha Namorada, Chistiane Dessana, pelo amor e carinho. Sua preocupação e força dada a todo o momento foram muito importantes, principalmente nos momentos mais difíceis.

Aos meus Avós, José Dirson, Arilo, Edite e Edith (em memória). Aos meus Tios, Abílio, Darlison, Dílson, Meire, Rose, Susana, Wálter, e Primos, Andrey, Caio, Gustavo, Renan, Tamires, Thiago, Weyden, Weyne, pela torcida e apoio constante.

Ao meu Orientador, Guilherme Travassos, pela grande dedicação, conselhos e motivação durante todo este período. Agradeço também pela orientação, incentivo e por me conduzir durante este trabalho e, por acreditar em mim e no meu trabalho.

Aos professores Ana Regina Rocha e José Carlos Maldonado por participarem de minha banca de defesa de mestrado.

Ao grande amigo Rafael, pela amizade incondicional ao longo deste período, por ter me aturado e me ajudado nos momentos mais complicados, e pela companhia a todo instante.

Especialmente aos amigos Gladys e Rodrigo, pelo incentivo desde a minha chegada ao Rio até o dia de hoje. Tenho certeza que levarei essas amizades para sempre.

Aos Companheiros da COPPE, Ana Cândida, Beto, Fernanda, Gleison, Luiz Gustavo, Hélio, Jóbson, Leonardo, Marco Antônio, Marcos, Mariano, Muradas, Paula, Paulo Sérgio, Reinaldo, Sávio, Sômulo, Taísa, Tayana e Wladmir, pela amizade, sugestões e ajuda nos momentos que precisei.

A outros amigos, Bruno, Caetano, Daniel, Eduardo, Gustavo, Jaysa e Roberta.

Aos amigos e professores da UFAM, especialmente aos professores Altigran e Rosiane, aos quais sempre serei grato.

Aos profissionais e diretoria das organizações do Estado do Amazonas que aceitaram o convite e participaram da pesquisa de opinião realizada neste trabalho.

À FAPEAM pelo apoio financeiro.

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

UMA INFRA-ESTRUTURA COMPUTACIONAL PARA APOIAR O PLANEJAMENTO E CONTROLE DE TESTES DE SOFTWARE

Arilo Claudio Dias Neto

Abril/2006

Orientador: Guilherme Horta Travassos

Programa: Engenharia de Sistemas e Computação

Teste de Software é uma das atividades mais custosas do processo de desenvolvimento e envolve muitos recursos, pessoas, prazos e riscos. Os principais problemas associados aos testes identificados na literatura estão relacionados a limitações nas atividades de planejamento e controle. No entanto, o sucesso desta atividade está completamente associado ao seu planejamento e controle.

Na literatura técnica da área de teste de software, pode ser encontrado um reduzido número de estudos sobre as atividades de planejamento e controle de testes de software, dificultando a caracterização do estado da arte desta área. Além disso, foi possível observar por meio de uma pesquisa de opinião realizada na indústria de software um grande interesse das organizações em aplicar práticas que apóiam o planejamento e controle de teste de software em seus projetos.

Desta forma, este trabalho apresenta: (1) descrição de abordagens que apóiam as atividades de planejamento e controle de testes de software, e suas características principais, identificadas na literatura técnica (estado da arte); (2) indícios, obtidos a partir de algumas organizações de software brasileiras, sobre as práticas de teste de software aplicadas em projetos (estado da prática); (3) proposta de abordagem para apoiar as atividades de planejamento e controle de teste de software representada por Maraká, uma infra-estrutura computacional que fornece apoio ao planejamento e controle de teste de software.

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

A COMPUTATIONAL INFRASTRUCTURE TO SUPPORT THE SOFTWARE TESTING PLANNING AND CONTROL

Arilo Claudio Dias Neto

April/2006

Advisor: Guilherme Horta Travassos

Department: Computer Science and Systems Engineering

Software Testing is one of the more expensive activities of the software development process. It involves a lot of resources, professionals, deadlines and risks. The main issues regarding software testing identified in the technical literature are related to the lack and limitation of the software testing planning and control activities. The success of testing can be related to its planning and control activities.

In the technical literature concerned with software testing, we can find a reduced number of studies about the planning and control of testing, what difficult the characterization of the state-of-the-art in this area. Moreover, we could observe by a survey executed in the Brazilian software industry the interest of software organizations in applying software testing practices to support the software testing planning and control activities in your projects.

Therefore, this work presents: (1) descriptions of approaches to support the software testing planning and control activities, and their main characteristics, identified in the technical literature (state-of-the-art); (2) insight, obtained from some Brazilian software organizations, regarding software testing practices applied in software projects (state-of-the-practice); (3) proposal of approach to support the software testing planning and control activities represented by Maraká, a computational infrastructure developed aiming at supporting the software testing planning and control activities.

Índice

Capítulo 1 Introdução	1
1.1 – Motivação	1
1.2 – Objetivo do Trabalho	5
1.3 – Metodologia de Trabalho	6
1.4 – Organização deste Trabalho	7
Capítulo 2 Abordagens de Apoio ao Planejamento e Controle de Testes de Software	9
2.1 – Introdução	9
2.2 – Definição da Taxonomia de Termos relacionados a Teste de Software	11
2.3 – Processo de Testes em Engenharia de Software	13
2.3.1 – Objetivos e Benefícios do Processo de Testes de Software	14
2.3.2 – Atividade de Teste em Modelos de Referência de Processo de Software ...	18
2.3.3 – Modelos de Referência de Processo de Testes de Software	24
2.4 – Utilização de Revisão Sistemática para Identificação de Abordagens para Apoiar o Planejamento e Controle de Testes	27
2.4.1 – Revisão Sistemática	27
2.4.2 – Protocolo de Revisão Sistemática para identificação das Abordagens.....	28
2.4.3 – Abordagens para Planejamento e Controle de Testes de Software	29
2.4.4 – Caracterização das Abordagens Identificadas	36
2.5 – Conclusão	39
Capítulo 3 Pesquisa de Opinião para Caracterização de Práticas de Teste em Organizações de Software	40
3.1 – Introdução	40
3.2 – Definição e Planejamento do Estudo	41
3.2.1 – Objetivo do Estudo	42
3.2.2 – Definição das Hipóteses	42
3.2.3 – Seleção das Variáveis	44
3.2.4 – Descrição da Instrumentação	44
3.2.5 – Contexto de Aplicação do Estudo	48
3.3 – Execução e Resultados do Estudo	49
3.3.1 – Cenário de Aplicação do Estudo	49
3.3.2 – Dados Obtidos	49
3.3.3 – Resultados do Estudo	51
3.4 – Avaliação da Aplicabilidade x Grau de Importância	64
3.4.1 – Aplicabilidade x Grau de Importância no Contexto Geral	66

3.4.2 – Aplicabilidade x Grau de Importância em Micros e Pequenas Empresas	69
3.4.3 – Aplicabilidade x Grau de Importância em Grandes Empresas	71
3.4.4 – Avaliação de Micros e Pequenas Empresas X Grandes Empresas	73
3.5 – Conclusão	73
Capítulo 4 Proposta de Apoio ao Planejamento e Controle de Testes de Software	76
4.1 – Introdução	76
4.2 – Descrição da Abordagem e seus Elementos	77
4.2.1 – Documentação dos Testes	77
4.2.2 – Sistematização dos Testes	78
4.3 – IEEE <i>Standard</i> 829: Padrão para Documentação dos Testes	79
4.4 – Definição do Processo de Testes de Software	81
4.4.1 – Requisitos do Processo de Testes de Software	82
4.4.2 – Composição do Processo de Testes de Software	85
4.5 – Conclusão	95
Capítulo 5 Maraká: Infra-estrutura Computacional para Apoio ao Planejamento e Controle de Testes de Software	97
5.1 – Introdução	97
5.2 – Requisitos da Infra-estrutura Maraká	98
5.2.1 – Apoio à Documentação e Sistematização dos Testes de Software	98
5.2.2 – Outros Requisitos da Infra-estrutura	102
5.3 – Arquitetura, Decisões de Projeto e Projeto de Alto Nível	102
5.3.1 – Solução Arquitetural	103
5.3.2 – Utilização do <i>Framework</i> Mambo	104
5.3.3 – Diagrama de Classes	106
5.4 – A Infra-estrutura Maraká	107
5.4.1 – <i>Login</i> e Menu Principal de Maraká	108
5.4.2 – Gerenciador da Equipe de Teste	109
5.4.3 – Gerenciador de Testes Instanciados	111
5.4.4 – Testes em Andamento	111
5.4.5 – Gerenciador do Processo de Testes Instanciado	112
5.4.6 – Alteração dos dados da conta de acesso à Maraká	118
5.5 – Conclusão	118
Capítulo 6 Avaliação de Maraká em relação aos itens da Pesquisa de Opinião	119
6.1 – Introdução	119
6.2 – Avaliação de Maraká	120
6.2.1 – Avaliação dos Itens Genéricos	120
6.2.2 – Avaliação dos Itens de Organização	121

6.2.3 – Avaliação dos Itens de Planejamento.....	124
6.2.4 – Avaliação dos Itens de Controle	127
6.2.5 – Avaliação dos Itens de Medição e Análise	131
6.2.6 – Avaliação dos Itens de Ferramenta	133
6.3 – Conclusão.....	135
Capítulo 7 Considerações Finais.....	138
7.1 – Visão Geral.....	138
7.2 – Contribuições.....	139
7.3 – Limitações do Trabalho	140
7.4 – Trabalhos Futuros.....	141
Referências Bibliográficas	143
Anexo A Notação usada na Modelagem do Processo de Testes de Software	149
Apêndice A Questionários Aplicados na Pesquisa de Opinião para Caracterização de Práticas de Revisão e Teste de Software.....	150
Apêndice B Dados Coletados na Pesquisa de Opinião para Caracterização de Práticas de Revisão e Teste de Software.....	151
Apêndice C Roteiros dos Documentos de Teste descritos no IEEE <i>Standard</i> 829.....	154

Capítulo 1

Introdução

Neste Capítulo são apresentadas as questões que levaram a realização deste trabalho, os objetivos a serem atingidos com ele, a metodologia seguida no seu desenvolvimento e a sua organização.

1.1 – Motivação

Teste de Software é uma das técnicas de verificação e validação de software (V&V) e consiste na análise dinâmica do produto, ou seja, na sua execução com o objetivo de provocar uma *falha* nesse produto, contribuindo para uma futura detecção de *defeitos* por meio de um processo de depuração e, conseqüentemente, o aumento da confiança de que o produto esteja correto (adaptado de ROCHA *et al.*, 2001).

Estas definições seguem a terminologia padrão para Engenharia de Software do IEEE – *Institute of Electrical and Electronics Engineers* – (IEEE 610, 1990).

- **Defeito (*fault*)** é um ato inconsistente cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta. Como por exemplo uma instrução ou comando incorreto.
- **Engano (*mistake*)** é uma ação humana que produz um resultado incorreto, como uma ação incorreta tomada pelo programador.
- **Erro (*error*)** é uma manifestação concreta de um defeito num artefato de software. Diferença entre o valor obtido e o valor esperado, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa constitui um erro.
- **Falha (*failure*)** é o comportamento operacional do software diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros e alguns erros podem nunca causar uma falha.

No processo de desenvolvimento de software todos os defeitos são humanos e apesar do uso dos melhores métodos de desenvolvimento, ferramentas ou profissionais, permanecem presentes nos produtos (HOWDEN,1987). Com isso, as atividades de teste possuem um papel fundamental no processo de desenvolvimento

de um software como mecanismo de apoio à garantia da qualidade do produto, pois corresponde ao último recurso para avaliação do produto antes da sua entrega ao usuário final (PRESSMAN, 2005).

Os testes devem ocorrer em diferentes níveis, de acordo com as fases do processo de desenvolvimento. Os principais níveis de teste de software, segundo (ROCHA *et al.*, 2001), são:

- *Teste de Unidade*: também conhecido como testes unitários. Tem por objetivo explorar a menor unidade do projeto, procurando provocar falhas ocasionadas por defeitos de lógica e de implementação em cada módulo, separadamente. O universo alvo desse tipo de teste são os métodos dos objetos ou mesmo pequenos trechos de código.
- *Teste de Integração*: visa a provocar falhas associadas às interfaces entre os módulos quando esses são integrados para construir a estrutura do software que foi estabelecida na fase de projeto.
- *Teste de Sistema*: avalia o software em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final. Dessa maneira, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia de manipulação do software. Verifica se o produto satisfaz seus requisitos.
- *Teste de Aceitação*: são realizados geralmente por um restrito grupo de usuários finais do sistema. Esses simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.

No entanto, apesar da sua eficiência contribuindo para o aumento da qualidade do produto final, as atividades de teste, assim como diversas outras atividades do processo de desenvolvimento de software, possuem problemas que precisam ser evitados durante a sua realização para garantir o seu sucesso. Segundo MATS (2001), os cinco principais problemas associados aos teste em software são:

1. Atrasos no cronograma do projeto, impossibilitando a equipe de teste de completar os testes planejados devido à redução de recursos e tempo;
2. Carência na rastreabilidade de casos e procedimentos de teste entre diferentes versões do software, dificultando a sua reutilização e repetição após modificações nos requisitos;
3. Teste manual ou não-padronizado, resultando em um grande esforço a cada início de uma nova atividade de teste;
4. Incerteza sobre o que está sendo testado, devido à falta de definição dos objetivos e escopo para as atividades de teste, e;

5. Ausência de critérios¹ para a seleção do conjunto de casos e procedimentos de testes², definição da sua completude e estabelecimento de um ponto de parada, dentre outros, dificultando a revelação de falhas no produto.

Analisando-se esses problemas descritos, observa-se a sua relação direta com a ausência ou limitação das atividades de planejamento e controle de testes de software.

No livro “Qualidade de Software: Teoria e Prática” (ROCHA *et al.*, 2001) é apresentado um questionamento que reflete exatamente as dificuldades e fatores de sucesso para as atividades de teste de software: “Uma vez conduzidos testes que não provocaram a falha no produto, o que se pode concluir? Que o produto em teste é de boa qualidade ou que os testes empregados são de baixa qualidade ou foram conduzidos sem planejamento, critérios e sem uma sistemática?”. No próprio livro é respondido que provavelmente a segunda opção é a mais verdadeira, devido às dificuldades em se desenvolver software sem a existência de defeitos.

Segundo BEIZER (1990) e JURISTO *et al.* (2004), teste de software é considerada uma das práticas mais custosa do processo de desenvolvimento, e desta forma, necessita de um bom gerenciamento a fim de evitar perdas de recursos e atrasos no cronograma, ou pior, não contribuir para a avaliação no produto, não ser concluída, dentre outras possibilidades. Com isso, a realização de testes em software requer recursos adequados, e a utilização efetiva desses recursos necessita de um bom planejamento e controle (MCGREGOR e SYKES, 2001). O planejamento garante que o critério de teste seja especificado e o conjunto de casos e procedimentos de testes seja desenvolvido antes do fim da fase de implementação do produto, e dessa forma evita testes tendenciosos, ou seja, testes que avaliam somente características do produto nas quais os desenvolvedores já tem conhecimento sobre o seu funcionamento. A atividade de controle assegura que os testes planejados sejam monitorados constantemente e seus resultados sejam registrados.

A realização de testes de software sem um planejamento pode ser comparada ao desenvolvimento de um projeto sem a criação de um plano para ele, e isto ocorre, normalmente, pelos mesmos motivos: grande pressão para iniciar a fase de execução o mais rápido possível (no caso de testes de software a execução está relacionada à aplicação dos casos de teste, e no desenvolvimento de software está relacionada à codificação) (MCGREGOR e SYKES, 2001).

¹ Um critério de teste é o que define quais propriedades precisam ser testadas para garantir inexistência de defeitos. Como é impossível garantir inexistência, o conceito é utilizado, na prática, para definir uma qualidade mínima que será avaliada pelo teste.

² Os termos “caso” e “procedimento” de teste estão descritos na Seção 2.1 do Capítulo 2.

Segundo BERTOLINO (2004), o cenário atual de teste de software apresenta a existência de uma grande distância entre o que é desenvolvido na comunidade acadêmica e o que é utilizado em organizações de desenvolvimento de software. As pesquisas na área de teste de software abordam principalmente estudos sobre técnicas de automatização dos testes que dificilmente chegam à indústria (FRANKL *et al.*, 1997; KIM *et al.*, 2000).

Apesar da notável importância do planejamento e controle para o sucesso dos testes, observa-se na literatura técnica da área de testes pouca importância dada a elas. Poucas abordagens que apoiem essas atividades são propostas, e normalmente para um contexto específico, e um número mais reduzido ainda é aplicado na indústria, caracterizando o estado da arte das atividades de planejamento e controle dos testes.

Nesse contexto, os modelos de referência de processo, como CMMI (2000) e mps.BR (2005), surgem como mecanismo para apoiar as atividades de planejamento e controle em diversos processos, dentre eles o processo de testes de software. A particularidade desses modelos é que eles envolvem o planejamento e controle de uma forma geral, não abordando características específicas das atividades de teste de software, como apoio à seleção do critério a ser adotado para a realização dos testes em um produto. Apesar disso, esses modelos devem servir como base para a realização do planejamento e controle dos testes em um projeto.

Isso dificulta a estruturação, o planejamento e uma monitoração constante dos testes realizados em uma organização.

A partir da dificuldade de caracterização do estado da arte das atividades de planejamento e controle de teste de software, devido à carência de estudos científicos sobre este tema, torna-se necessário obter um entendimento de como estas atividades estão sendo aplicadas em ambientes reais de desenvolvimento de software para que a partir das reais necessidades de organizações de software, novas estratégias possam ser adotadas.

Com o intuito de iniciar uma caracterização do cenário atual com relação às atividades de teste de software em um cenário de desenvolvimento de software, foi realizada uma pesquisa de opinião em organizações de software, localizadas em um pólo de desenvolvimento de software brasileiro, cujo objetivo era a avaliação de quais práticas³ de teste de software estão sendo utilizadas nesse ambiente real de desenvolvimento. Esta pesquisa será descrita detalhadamente ao longo deste trabalho. Os resultados indicaram uma carência e limitação na aplicação de práticas

³ As práticas de teste de software avaliadas correspondem a um conjunto de atividades ou procedimentos de apoio à realização dos testes.

relacionadas ao planejamento e controle de teste de software nessas organizações que participaram da pesquisa, porém observou-se o interesse dessas organizações em relação a elas.

Baseado neste cenário, este trabalho apresenta uma nova proposta de apoio ao planejamento e controle de testes de software, por meio da construção de uma infra-estrutura computacional que possibilite a monitoração e documentação de testes de software.

1.2 – Objetivo do Trabalho

A partir da importância das atividades de teste de software e da necessidade de um planejamento e controle como mecanismo para garantir o seu sucesso, aliado ao interesse das organizações na aplicação de práticas que apóiam essas atividades, conforme descrito anteriormente, este trabalho tem como principal objetivo a elaboração e disponibilização de uma infra-estrutura computacional que apóia o planejamento e controle de testes de software, fornecendo mecanismos para acompanhamento do processo de testes de software e a documentação dos testes realizados em organizações de software.

Para fornecer o apoio adequado às atividades de planejamento e controle de testes de software, os requisitos para a infra-estrutura foram identificados a partir de informações descritas na literatura técnica relacionada à área de teste de software e de observações extraídas de uma pesquisa de opinião realizada em organizações de software brasileiras localizadas em um cenário específico, cujo objetivo foi obter uma caracterização inicial do estado da prática das atividades de teste em organizações que desenvolvem software.

Com isso, a infra-estrutura, denominada Maraká, foi projetada e construída, fornecendo um apoio automatizado ao planejamento e controle de testes de software. Entre as funcionalidades fornecidas por Maraká estão o gerenciamento da equipe de testes de software e sua alocação nas atividades de teste em um projeto e a execução e acompanhamento do processo de testes de software e seus artefatos.

Com isso, existem duas expectativas principais para Maraká: (1) a sua utilização em atividades de teste de software, possibilitando a realização de um planejamento e controle dos testes de forma mais sistematizada por organizações de software e (2) a documentação dos testes de software seguindo padrões internacionais e processo pré-estabelecidos, fornecendo um mecanismo inicial para

adequação das atividades de teste em organizações de software a modelos de referência de processos de software, como CMMI (2000) ou MPS.BR (MPS.BR, 2005).

Visando à avaliação inicial do conjunto de funcionalidades disponibilizadas pela infra-estrutura Maraká na realização das atividades de planejamento e controle dos testes de software, foi analisado o apoio fornecido por Maraká a cada uma das práticas de teste de software que compôs a pesquisa de opinião executada em organizações de software e que faz parte deste trabalho.

1.3 – Metodologia de Trabalho

A metodologia utilizada para a realização deste trabalho seguiu os seguintes passos:

- Estudo sobre o processo de testes de software, enfatizando as atividades de planejamento e controle dos testes, detalhando seus conceitos, suas características e benefícios para uma organização de software;
- Identificação e caracterização de abordagens científicas que apóiam o planejamento e controle de testes de software;
- Planejamento e execução de uma pesquisa de opinião em um cenário de desenvolvimento de software brasileiro com o objetivo de extrair observações relacionadas ao estado atual de práticas de teste de software em ambientes reais de desenvolvimento. Com isso, foram observados indícios a respeito das carências e limitações das atividades de planejamento e controle de testes em organizações de software;
- Definição de uma abordagem para apoiar o planejamento e controle dos testes, composta por dois elementos básicos: **documentação** e **sistematização**, apresentada em (DIAS NETO e TRAVASSOS, 2005a);
 - Definição do mecanismo de apoio à documentação dos testes a ser utilizado no trabalho: IEEE *Standard 829* (1998);
 - Definição de um processo de testes de software a ser utilizado ao longo do trabalho, como mecanismo adotado para a sistematização dos testes;
- Definição dos requisitos funcionais e não-funcionais visando à construção de uma infra-estrutura computacional, proposta em (DIAS NETO e TRAVASSOS, 2005b), que apóia o planejamento e controle dos testes de software por meio da aplicação da abordagem definida no passo anterior, possibilitando a documentação e sistematização das atividades de teste. Os requisitos foram obtidos a partir de duas fontes de conhecimento: a

literatura da área de teste (visão científica) e os resultados obtidos na pesquisa de opinião realizada (visão prática);

- Definição da solução arquitetural e do projeto de alto nível da infraestrutura computacional, viabilizando o seu desenvolvimento;
- Construção e avaliação da infra-estrutura computacional proposta, denominada Maraká, visando ao apoio às atividades de planejamento e controle de testes de software segundo os requisitos funcionais e não-funcionais definidos, e;
- Escrita de monografias, relatórios técnicos, artigos a serem publicados em eventos científicos e da dissertação de mestrado ao longo do trabalho.

1.4 – Organização deste Trabalho

Além desta introdução, este trabalho é composto por outros seis capítulos, um anexo e três apêndices.

No Capítulo 2, *Abordagens de Apoio ao Planejamento e Controle de Testes de Software*, são definidos os principais conceitos relacionados ao processo de testes de software, suas características e benefícios para uma organização de software, ressaltando as atividades de planejamento e controle dos testes. Ainda neste capítulo, são apresentadas e caracterizadas abordagens que apóiam o planejamento e controle dos testes de software identificadas na literatura, apresentando o estado da arte com relação às atividades de planejamento e controle de teste de software.

No Capítulo 3, *Pesquisa de Opinião para Caracterização de Práticas de Teste em Organizações de Software*, são apresentados o planejamento, execução e os resultados de uma pesquisa de opinião realizada em organizações localizadas em um cenário de desenvolvimento de software brasileiro. Esta pesquisa possui o objetivo de obter uma caracterização inicial de quais práticas de teste estão sendo utilizadas em ambientes reais de desenvolvimento. Dessa forma, busca-se identificar eventuais carências nas atividades de teste *in loco*, e a partir desses resultados prover mecanismos que possam contribuir para a realização das atividades de teste.

No Capítulo 4, *Proposta de Apoio ao Planejamento e Controle de Testes de Software*, é apresentada uma abordagem definida para apoiar o planejamento e controle dos testes de software a ser utilizada ao longo deste trabalho. Neste Capítulo são apresentados os dois elementos que compõem a abordagem proposta: o IEEE *Standard 829*, que será utilizado como mecanismo para documentação dos testes ao longo do trabalho, e o processo de testes de software, com suas atividades, critérios

de entrada e saída em cada atividade, artefatos e papéis associados, definido neste trabalho e que será utilizado como mecanismo para sistematização dos testes.

No Capítulo 5, *Maraká: Infra-estrutura Computacional para Apoio ao Planejamento e Controle de Testes de Software*, é apresentada a solução computacional para aplicação da abordagem proposta no Capítulo 4. São descritos os requisitos da infra-estrutura, sua solução arquitetural, projeto de alto nível e as decisões de projeto realizadas. Por fim, é apresentada a infra-estrutura Maraká por meio da descrição de suas funcionalidades, visando ao apoio às atividades planejamento e controle dos testes de software.

No Capítulo 6, *Avaliação de Maraká em relação aos itens da Pesquisa de Opinião*, é descrita uma avaliação da infra-estrutura Maraká em relação ao apoio fornecido por Maraká para a realização de cada prática de teste de software que compôs a pesquisa de opinião para caracterização da atividade de teste em ambientes reais de desenvolvimento, apresentada no Capítulo 3 deste trabalho.

No Capítulo 7, *Considerações Finais*, são apresentadas as conclusões e os resultados obtidos ao longo deste trabalho, suas principais contribuições e limitações, e os próximos passos a serem realizados dando continuidade a este trabalho.

No Anexo A, *Notação usada na Modelagem do Processo de Testes de Software*, é apresentada a notação gráfica utilizada para a modelagem do processo de testes de software definido neste trabalho. Esta notação está descrita completamente em (VILLELA, 2004).

No Apêndice A, *Questionários Aplicados na Pesquisa de Opinião para Caracterização de Práticas de Revisão e Teste de Software*, são apresentados os questionários utilizados na pesquisa de opinião realizada neste trabalho, cujo objetivo foi avaliar o estado da prática das atividades de teste de software em um cenário de desenvolvimento de software brasileiro, descrita no Capítulo 3.

No Apêndice B, *Dados Coletados na Pesquisa de Opinião para Caracterização de Práticas de Revisão e Teste de Software*, são apresentados todos os dados coletados durante a execução da pesquisa de opinião realizada neste trabalho, porém é mantida a confidencialidade dos dados.

Por fim, no Apêndice C, *Roteiros dos Documentos de Teste descritos no IEEE Standard 829*, são apresentados os roteiros para preenchimento dos documentos de teste descritos no IEEE Standard 829 (1998) e que serão utilizados neste trabalho como artefatos produzidos ao longo do processo de testes de software.

Capítulo 2

Abordagens de Apoio ao Planejamento e Controle de Testes de Software

Neste capítulo são apresentadas as características e os benefícios do planejamento e controle de testes de software, que correspondem a um conjunto de atividades que fazem parte do processo de testes de software e são essenciais para o seu sucesso. Além disso, são caracterizadas abordagens para planejamento e controle de testes de software indicando seus pontos positivos e negativos, representando o estado da arte das atividades de planejamento e controle de teste de software.

2.1 – Introdução

Planejamento e controle são tarefas essenciais na execução de qualquer atividade que envolva muitas pessoas, recursos adequados, prazos definidos e riscos associados. A atividade de planejamento permite a definição prévia das ações a serem realizadas na execução de uma tarefa específica, enquanto que a atividade de controle possibilita que as ações planejadas sejam monitoradas constantemente durante a sua execução e que novas ações possam ser tomadas a partir do que foi realizado efetivamente.

Segundo PRESSMAN (2005), a gerência de projetos de software envolve planejamento, acompanhamento e controle de pessoas, processos e eventos que ocorrem à medida que o software evolui de um conceito inicial a uma implementação operacional.

Para a realização de um projeto, é necessário que haja o planejamento de quais atividades deverão ser realizadas, como elas deverão ser executadas e qual o contexto em que elas estão inseridas. O planejamento e controle de um projeto em Engenharia de Software englobam o planejamento e controle do processo de desenvolvimento do software, recursos humanos, hardwares e softwares necessários, custos relacionados, cronograma, qualidade do produto, documentação e riscos envolvidos.

O Instituto de Gerência de Projetos (*Project Management Institute - PMI*) descreve em um documento – *PMBOK: Project Management Body Of Knowledge* – (2004) a prática de Gerência de Projeto e suas áreas de conhecimento (Gerência de custos, tempo, escopo, recursos humanos, riscos, qualidade e comunicação) que são fundamentais no desenvolvimento de um projeto. Este documento ressalta a importância das atividades de planejamento e controle para o gerenciamento de cada uma dessas áreas de conhecimento.

Em Engenharia de Software, observa-se a constante aplicação das atividades de planejamento e controle durante algumas atividades que compõem o processo de desenvolvimento de um software, como planejamento de riscos (FARIAS, 2002), de custos (BARCELLOS, 2003) ou de recursos humanos (SCHNAIDER, 2003). Isso motiva a aplicação de planejamento e controle para outras atividades do processo de desenvolvimento, como teste de software. Essas atividades são ressaltadas pela existência de um padrão internacional para documentação dos testes de software: *IEEE Standard 829 (1998)*.

Segundo BERTOLINO (2004), as estratégias que apóiam o planejamento e controle de testes de software aplicadas na indústria se refletem, atualmente, em tentativas individuais de implantar ou mesmo melhorar o processo de testes utilizado. No entanto, há uma grande dificuldade por parte das organizações de software em entender o que é necessário para a realização do planejamento e controle dos testes, principalmente relacionada à escolha das técnicas de teste⁴ a serem utilizadas. Ainda que as técnicas de teste de software mais utilizadas foram criadas por volta dos anos 70, as organizações têm grandes dificuldades em aplicá-las devido à sua complexidade e ausência de conhecimento científico sobre as técnicas, indicando em quais contextos elas podem ser aplicadas e quais os resultados da sua utilização, comparando-as com outras técnicas.

Nesse contexto, neste Capítulo são identificadas abordagens que apóiem o planejamento e controle de testes de software na literatura técnica como uma forma de se avaliar o estado da arte nesta área.

Sendo assim, na Seção seguinte são apresentadas as definições de alguns conceitos relacionados às atividades de teste de software e que serão utilizados ao longo deste trabalho. Na Seção 2.3 são apresentados os conceitos relacionados ao processo de testes de software, os seus objetivos, os benefícios oferecidos para uma organização com a sua utilização, a sua relação com o processo de desenvolvimento de software e onde as atividades de planejamento e controle dos testes estão inseridas dentro desse processo.

⁴ Ver Seção 2.1.

Na seção 2.4 são apresentados os principais conceitos relacionados à revisão sistemática (KITCHENHAM, 2004), a descrição do protocolo de revisão sistemática utilizado para identificação das abordagens que apóiem o planejamento e controle de testes de software, a execução da revisão sistemática e uma caracterização das abordagens identificadas, comparando-as a partir de critérios previamente definidos. Por fim, na seção 2.5 são descritas as conclusões deste capítulo.

2.2 – Definição da Taxonomia de Termos relacionados a Teste de Software

Nessa seção são definidos os principais termos utilizados neste trabalho para que sejam evitadas interpretações ambíguas e facilite o seu entendimento.

Sendo assim, Teste de Software, no contexto deste trabalho, é definido como uma técnica de verificação e validação que analisa dinamicamente um produto desenvolvido ao longo de um processo de desenvolvimento de um software e que possui o objetivo de provocar *falhas* em um produto, para que a partir de um processo de *depuração* sejam identificados *defeitos* existentes neste produto (adaptado de ROCHA *et al.*, 2001), (MCGREGOR e SYKES, 2001).

O processo de depuração é o processo de detecção e remoção ou correção de um defeito identificado no software.

Ao longo deste trabalho, outros termos serão utilizados e precisam ser definidos. Os termos são:

- **Caso de Teste.** Descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado (CRAIG e JASKIEL, 2002).
- **Procedimento de Teste.** É uma descrição dos passos necessários para executar um caso (ou um grupo de casos) de teste (CRAIG e JASKIEL, 2002).
- **Critério de Teste:** Um critério de teste serve para selecionar e avaliar casos de teste de forma a aumentar as possibilidades de provocar falhas ou, quando isso não ocorre, estabelecer um nível elevado de confiança na correção do produto (ROCHA *et al.*, 2001). Os critérios de teste podem ser utilizados como:
 - **Critério de Cobertura dos Testes:** Os critérios de cobertura para o teste de software permitem a identificação de partes do programa que devem ser executadas para garantir a qualidade do software e

indicar quando o mesmo foi suficientemente testado (RAPPS e WEYUKER, 1982). Ou seja, determinar o percentual de elementos necessários por um critério de teste que foram executados pelo conjunto de casos de teste (ROCHA *et al.*, 2001).

- **Critério de Adequação de Casos de Teste:** Quando, a partir de um conjunto de casos de teste T qualquer, ele é utilizado para verificar se T satisfaz os requisitos de teste estabelecidos pelo critério (ROCHA *et al.*, 2001).
- **Critério de Geração de Casos de Teste:** quando o critério é utilizado para gerar um conjunto de casos de teste T adequado por construção ao critério (ROCHA *et al.*, 2001).
- **Técnica de Teste:** As Técnicas de Teste são classificadas de acordo com a origem das informações utilizadas para estabelecer os requisitos de teste. Elas contemplam diferentes perspectivas do software e impõe-se a necessidade de se estabelecer uma estratégia de teste que contemple as vantagens e os aspectos complementares dessas técnicas. As técnicas existentes são: técnica funcional, estrutural, com base em defeitos, com base em máquinas de estado finito, dentre outras. Cada uma das técnicas de teste mencionadas possui diversos critérios de teste, como por exemplo:
 - **Técnica Funcional:** Particionamento de equivalência, análise do valor limite e grafo causa-efeito.
 - **Técnica Estrutural:** critérios com base na complexidade, critérios com base em fluxo de controle e critérios com base no fluxo de dados.
 - **Técnica com base em defeitos:** Semeadura de defeitos, análise de mutantes.

O termo Processo de Testes de Software, neste trabalho, está relacionado a um conjunto sistematizado de passos, composto por atividades, papéis e artefatos, necessários para a realização do planejamento, controle e execução dos testes ao longo do processo de desenvolvimento (CRAIG e JASKIEL, 2002). O processo de testes de software guia a identificação, especificação e execução de casos e procedimentos de teste.

Ao longo deste trabalho, o termo “teste de software” será chamado por diversas vezes simplesmente de “teste”, para simplificar a sua escrita e o seu entendimento. O

mesmo irá ocorrer com o termo “processo de desenvolvimento de software”, que será tratado como “processo de desenvolvimento”, somente.

2.3 – Processo de Testes em Engenharia de Software

Um processo, no contexto de Engenharia de Software, consiste em uma série de passos envolvendo atividades, restrições e recursos que produzem um resultado esperado (PFLEEGER, 2004). Segundo a ontologia de processo no contexto de Engenharia de Software apresentada em (FALBO, 1998), um processo é uma coleção de atividades relacionadas que têm lugar definido durante o desenvolvimento de um produto.

Processos são importantes porque impõem consistência e estrutura a um conjunto de atividades, guiam ações e permitem examinar, entender, controlar e melhorar as atividades que o compõem. As principais características de um processo, no contexto de Engenharia de Software, segundo PFLEEGER (2004), são:

- Utilização de recursos previamente estabelecidos e sujeitos a um conjunto de restrições (como cronograma ou orçamento);
- Desenvolvimento, ao longo de suas atividades, de produtos intermediários e finais, que são os artefatos;
- Podem ser compostos por sub-processos;
- Cada atividade do processo possui um critério de entrada (início) e saída (fim), e;

A norma ISO/IEC 12207 (1995) descreve um conjunto de processos fundamentais, organizacionais e de apoio que compõem o ciclo de vida de um software, entre os quais está o processo de desenvolvimento. Este processo é responsável pelas atividades do desenvolvedor (segundo esta norma, o desenvolvedor é a organização que define e desenvolve o produto de software). Isso inclui as atividades de análise de requisitos, projeto, codificação, integração e testes em um produto.

Os testes em software devem ser planejados e controlados durante todo o processo de desenvolvimento a fim de evitar perdas de recursos ou atrasos no cronograma, mantendo a qualidade desta atividade. Durante a realização dos testes, são elaborados diversos documentos, como Plano de Teste, Casos de Teste e Relatório de Incidentes detectados, que contêm informações sobre os testes realizados.

Apesar da importância das atividades de teste durante o desenvolvimento de software, testes de software normalmente são vistos pelos profissionais da área como sendo uma simples atividade do processo de desenvolvimento realizada após a codificação do produto (BRYAN e SIEGEL, 1988). No entanto, apesar de fazer parte do processo de desenvolvimento, alguns autores (GELPERIN e HETZEL, 1988; PERRY, 1995; BURNSTEIN e SUWANASSART, 1996; ABRAN *et al.*, 2001) reconhecem a importância e complexidade das atividades de teste e descrevem-nas como sendo um processo separado, porém integrado ao processo de desenvolvimento. Isto ocorre devido à diversos fatores, tais como: complexidade das atividades de teste, riscos e recursos envolvidos na sua execução e as características específicas deste processo, como será descrito a seguir.

2.3.1 – Objetivos e Benefícios do Processo de Testes de Software

Os objetivos do processo de testes de software são diferenciados em relação aos objetivos do processo de desenvolvimento, principalmente por possuírem medidas de sucesso diferentes. Enquanto o processo de desenvolvimento busca a construção de um produto com qualidade, o processo de testes almeja expor as falhas existentes nesse produto. No entanto, esses processos possuem algo em comum: o desejo de, ao final, obter um produto com qualidade. Um processo de testes possui características próprias em relação ao processo de desenvolvimento dentro de um projeto de software (MCGREGOR e SYKES, 2001). Essas características são:

- **Objetivo do processo:** o processo de testes busca responder questões sobre o produto, incluindo a análise de se o produto atende às necessidades para as quais foi idealizado, enquanto o processo de desenvolvimento busca construir um produto que atenda uma necessidade específica.
- **Natureza destrutiva e não construtiva:** o sucesso do processo de testes indica, de certa forma, o fracasso do processo de desenvolvimento. Um processo de testes está sempre “questionando” a qualidade do produto, tentando provocar falhas, e não é uma atividade de construção, por isso é chamado de “destrutivo”. Por exemplo, na perspectiva dos testadores, provocar falhas no produto representa atingir o seu objetivo, enquanto que para os desenvolvedores representa esforço adicional, apesar da consciência de que isso contribui para a qualidade do produto final.

- **Papéis e responsabilidades específicos:** os papéis definidos para os processos de testes e desenvolvimento são associados a diferentes perfis de profissionais, resultando na necessidade de habilidades e conhecimento distintos entre as pessoas envolvidas nesses processos. Os papéis associados ao processo de testes devem se preocupar em selecionar técnicas de teste e desenvolver casos de teste para fazer com que o sistema falhe, enquanto que os papéis associados ao processo de desenvolvimento devem se preocupar em modelar o sistema e construir linhas de código para ele. No entanto, os profissionais em computação são, normalmente, ensinados em cursos técnicos ou ensino superior a “construir” programas, e não a “destruí-los”, o que torna necessário um treinamento específico para os profissionais que atuam na área de teste de software.
- **Dependência do processo de desenvolvimento:** o processo de desenvolvimento é auto-contido, ou seja, tem acesso às informações necessárias à execução de suas atividades. Já o processo de testes está sempre associado a um processo de desenvolvimento, o que consiste em uma dependência. Cada etapa do processo de desenvolvimento pode ter uma atividade do processo de testes associada. Esse comportamento pode ser observado em diversos processos que compõem o desenvolvimento de um produto, como processo de Gerência de Requisitos, Gerência de Configuração, Monitoração e Controle, dentre outros.
- **Ciclos de interação com o processo de desenvolvimento:** Os processos de testes e de desenvolvimento devem estar separados, porém estão interligados em um ciclo de realimentação constante (Figura 2.1). O processo de testes é responsável pela geração de um conjunto de casos e procedimentos de teste a partir dos documentos de análise e projeto gerado no processo de desenvolvimento. E ao final, tenta provocar falhas no software originadas no processo de desenvolvimento. A partir das falhas reveladas, os desenvolvedores tentam identificar a localização exata do defeito. Com isso, novamente serão seguidas atividades do processo de desenvolvimento para tentar corrigir este defeito identificado, e este ciclo é seguido até a conclusão do produto.

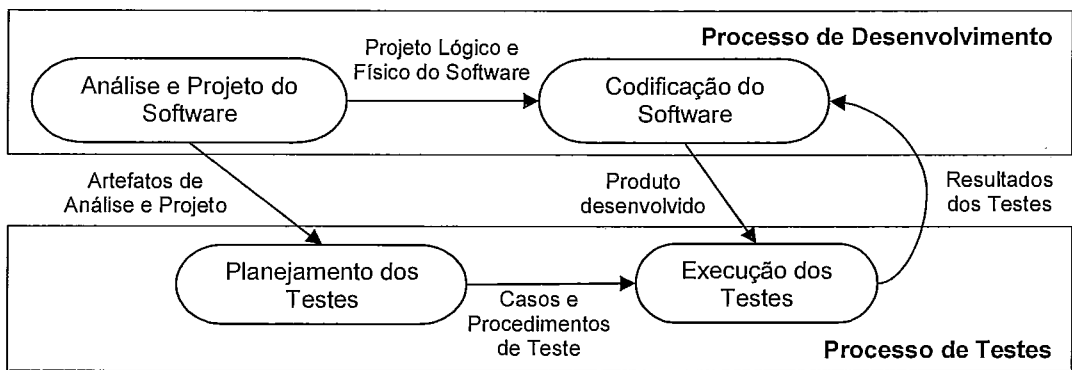


Figura 2.1. Ciclo entre o processo de desenvolvimento e de testes

A utilização de um processo para a realização de testes em software tem como objetivo a execução de diversas atividades que, segundo MCGREGOR e SYKES (2001), podem ser agrupadas em seis dimensões que serão descritas em forma de questões a serem respondidas:

- Quais partes do software serão testadas? Todos os componentes do software serão testados ou somente partes críticas e de alto risco?
- Como os testes serão realizados? Quais os critérios e técnicas para realização dos testes?
- Quem realizará os testes? Quem será responsável pelo planejamento dos testes? E pela execução dos testes?
- Quando os testes serão realizados? Em que momento do processo de desenvolvimento os testes serão executados?
- Onde os testes serão realizados? Em qual ambiente e qual a configuração do ambiente os testes ocorrerão?
- Qual a quantidade de testes adequada? Como decidir o que testar e quando finalizar os testes considerando os recursos limitados para a atividade?

Conforme citado anteriormente, a execução do processo de testes deve ocorrer ao longo do processo de desenvolvimento de um software, e possui diferentes tarefas de acordo com a atividade atual do processo (PERRY, 1995). O modelo-V (Figura 2.2) descreve a estrutura de um processo de desenvolvimento de software e o seu relacionamento com as atividades de teste.

Segundo este modelo, no momento em que o processo de desenvolvimento está na fase de especificação de requisitos, o planejamento dos testes de aceitação, que visa avaliar se os requisitos definidos pelo cliente estão atendidos no software, pode ser realizado. Na fase de projeto de alto nível, os testes de sistemas, que avaliam as funcionalidades do software, podem ser planejados. Durante a fase de

projeto detalhado, os testes de integração, que avaliam a integração entre os módulos ou componentes do software, podem ser planejados. Por fim, durante a fase de codificação, os testes de unidade, que avaliam a corretude das unidades (que podem ser classes ou procedimentos) do software, podem ser planejados (Figura 2.2).

No entanto, a execução dos testes normalmente ocorre na ordem inversa ao planejamento. Após a fase de codificação do software, deve ser iniciada a execução dos testes de unidade. Em seguida devem ser executados os testes de integração, testes de sistema e, por fim, testes de aceitação seguindo esta ordem.

O planejamento dos testes deve fazer parte do planejamento global do software, sendo o responsável pelo planejamento do processo de testes (por meio da descrição de prazos, recursos e custos estimados) e pela definição de estratégias, critérios e técnicas de teste a serem adotados para definição do conjunto de casos e procedimentos de teste necessários para avaliar o software.

O controle dos testes é responsável por garantir que as informações e atividades descritas durante o planejamento sejam realizadas conforme o planejado, e em caso de mudanças, que elas sejam relatadas às pessoas envolvidas nas atividades de teste. O controle dos testes também é responsável pelo registro dos eventos e incidentes ocorridos durante os testes e, com isso, possibilita a re-execução dos testes após modificações no produto.

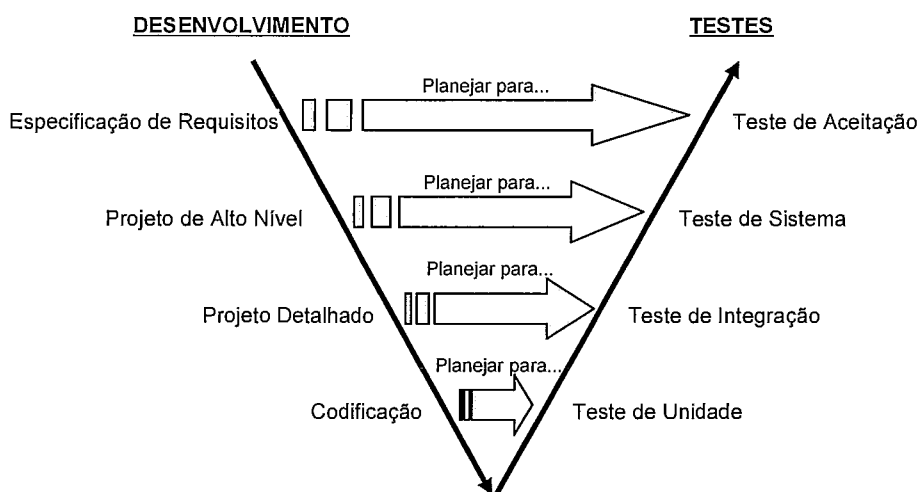


Figura 2.2 – Modelo-V (CRAIG e JASKIEL, 2002)

A realização de testes de software sem um planejamento pode ser comparada ao desenvolvimento de um projeto sem a criação de um plano para ele, e isto ocorre, normalmente, pelos mesmos motivos: grande pressão para iniciar a fase de execução o mais rápido possível (no caso de testes de software a execução está relacionada à aplicação dos casos e procedimentos de teste, e no desenvolvimento de software está relacionada à codificação) (MCGREGOR e SYKES, 2001).

A realização do planejamento e controle dos testes apresenta diversos benefícios para uma organização. Entre eles, pode-se citar:

- Papéis e responsabilidades claramente definidos, facilitando o gerenciamento dessa atividade e conseqüentemente as estimativas de custo, prazo e esforço;
- Objetivos dos testes explícitos, simplificando o entendimento sobre os diferentes tipos de teste que podem ocorrer em um sistema e suas razões, além de auxiliar a identificação de possíveis riscos para a atividade;
- Documentação dos testes bem definida, permitindo uma padronização entre os documentos e facilitando a leitura desses documentos em diferentes projetos, e;
- Facilidade de comunicação entre a equipe de testes e de desenvolvimento, criando um meio de comunicação formal entre essas equipes, definindo assim uma linguagem comum entre elas.

Desta forma, ressalta-se a importância do planejamento e controle dos testes durante o desenvolvimento de software. Na seção seguinte é discutido como os principais modelos de referência de processo de software abordam teste de software.

2.3.2 – Atividade de Teste em Modelos de Referência de Processo de Software

Diferentes modelos para avaliação de processos no contexto de Engenharia de Software são utilizados atualmente por organizações que buscam melhoria para seus processos. Estes modelos representam uma forma de atingir melhorias no desenvolvimento de software por meio de avaliação dos processos utilizados pela organização, comparando com um modelo de referência.

Neste sentido, será apresentada uma descrição de como os principais modelos de processo abordam questões relacionadas às atividades de teste de software. Os modelos apresentados são: CMM (SEI, 1993), ISO/IEC 12207 (1995), ISO/IEC 15504 (2003), CMMI (2000) e MPS *BR* (2005).

2.3.2.1 – Teste de Software de acordo com o CMM

O modelo CMM – *Capability Maturity Model* – (SEI, 1993) foi desenvolvido pelo SEI (*Software Engineering Institute*) como mecanismo de avaliação do processo de desenvolvimento de organizações, a fim de prover melhorias e constante evolução no processo. O CMM define cinco níveis de maturidade, onde cada nível indica um grau

de maturidade dos processos da organização: Inicial, Repetível, Definido, Gerenciado e Otimizado.

No livro *Systematic Software Testing*, CRAIG e JASKIEL (2002) descrevem a forma como o modelo CMM aborda questões relativas a testes de software. Os autores destacam que entre as 17 áreas chave de processo (em inglês *KPA*), não há uma específica para Teste, que está diluído em outras *KPAs*. Segundo os autores, o CMM até o nível 3 (Definido) não aborda as atividades de teste.

No nível 2 (Repetível), o gerente de teste deve se empenhar para determinar objetivos dos testes específicos e iniciar um processo de planejamento dos testes para a organização (CRAIG e JASKIEL, 2002). Isso ocorre por meio das atividades definidas na *KPA Planejamento de Projeto de Software*.

No nível 3, um processo para planejamento dos testes de software formal é introduzido para verificar quais requisitos foram satisfeitos em cada fase do ciclo de vida do desenvolvimento de software. Ainda no nível 3, o gerente de teste deve se empenhar para estabelecer uma organização de testes, ou seja, um grupo de testes dentro da organização. Os planos de teste devem ser integrados no ciclo de vida de desenvolvimento de software e o processo de testes deve ser controlado e monitorado. Esses aspectos identificados estão presentes nas atividades definidas na *KPA Engenharia do Produto de Software*.

No nível 4, o gerente de teste deve se empenhar para estabelecer um programa dentro da organização para revisão de software, treinamento técnico, medição dos testes e avaliação da qualidade de software, conforme descrito nas atividades definidas na *KPA Gerenciamento de Processo Quantitativo*.

Finalmente no nível 5, o gerente de teste deve se empenhar para aplicar controle de processo para prevenir defeitos futuros e dar ênfase nas atividades de controle de qualidade, conforme descrito nas atividades definidas na *KPA Prevenção de Defeitos*.

No entanto, o CMM não descreve suas práticas por meio de processos, o que ainda torna necessária a definição de um processo de testes para a utilização dos conhecimentos definidos por meio deste modelo.

2.3.2.2 – Teste de Software de acordo com a norma ISO/IEC 12207

A norma ISO/IEC 12207 (1995) tem como objetivo auxiliar os envolvidos na produção de software a definir seus papéis, por meio de processos bem definidos, facilitando o entendimento das atividades realizadas (ROCHA *et al.*, 2001).

No livro “Qualidade de Software – Teoria e Prática”, ROCHA *et al.* (2001) descrevem como as atividades de teste de software estão inseridas na norma ISO/IEC 12207. Segundo os autores, a atividade de teste de software é abordada nos processos de verificação e validação (V&V). O processo de validação é responsável pela preparação dos requisitos de teste, casos de teste e especificações de teste para analisar os seus resultados, condução dos testes com os usuários e validar se o produto de software satisfaz seu uso específico, testando-o no seu ambiente-alvo.

Além disso, teste de software é visto no contexto da norma também como parte do processo primário do ciclo de vida. A atividade de teste está inserida nos processos fundamentais do ciclo de vida da norma, mais especificamente no processo de desenvolvimento.

A primeira atividade relacionada ao teste de software é a de codificação e teste de software, que tem como objetivo testar cada unidade do software. As próximas atividades no processo de desenvolvimento que envolvem testes são integração do software e teste de qualificação do software. Finalmente, as outras atividades do processo de desenvolvimento relacionadas a teste de software são integração do sistema.

Além do processo de desenvolvimento, as atividades de teste estão presentes no processo de operação da norma, que trata de novas versões do software. Neste caso, os testes devem ser repetidos para garantir que tanto as partes alteradas quanto as partes inalteradas estejam funcionando adequadamente (Teste de Regressão).

2.3.2.3 – Teste de Software de acordo com a norma ISO/IEC 15504

ISO/IEC TR 15504 (2003) é o padrão internacional para Avaliação de Processo de Software. Ele foi publicado inicialmente em 1998 como um Relatório Técnico. ISO/IEC TR 15504 provê um *framework* para a avaliação de processos de software. Este *framework* pode ser usado por organizações e envolvem questões relacionadas ao planejamento, gerenciamento, monitoração, controle e melhoria da aquisição, fornecimento, desenvolvimento, operação, evolução e apoio ao software (ISO/IEC 15504-5, 2004).

Esta norma é um esforço independente da ISO/IEC. Ela foi inspirada pelo Software CMM e ISO 9001, mas o propósito da norma ISO/IEC 15504 é harmonizar modelos diferentes (incluindo o Software CMM, CMMI, ISO 9001, ISO 12207 e Bootstrap) e métodos de avaliação (incluindo CBA IPI, SCE 3.0, Bootstrap e SCAMPI) (SEI/CMU, 2006).

ISO/IEC TR 15504 contém um modelo de referência. Este modelo de referência define um modelo bidimensional de capacitação de processo: a *dimensão de processo* e a *dimensão da capacitação* (capability).

A *dimensão de processo* contém um conjunto de processos divididos em categorias, que são: *Ciclo de Vida Primário*, *Ciclo de Vida Organizacional*, *Ciclo de Vida de Suporte*. Em cada categoria, os processos são agrupados em um segundo nível de acordo com o tipo de atividade que eles desempenham (ISO/IEC 15504-5, 2004).

As atividades de teste estão presentes em duas categorias de processos: Ciclo de Vida Primário e Ciclo de Vida de Suporte.

Na categoria Ciclo de Vida Primário, testes estão presentes no Grupo de Processos de Engenharia – ENG (processos responsáveis por elicitar e gerenciar os requisitos do cliente, especificar, implementar e manter o produto de software) através de quatro processos: ENG 7 (Integração do Software), ENG 8 (Teste do Software), ENG 9 (Integração do Sistema), ENG 10 (Teste do Sistema).

Na categoria Ciclo de Vida de Suporte, testes estão presentes no Grupo de Processos de Garantia da Qualidade – QUA (executados para garantir que os produtos de trabalho e processos estão em conformidade com o que foi definidos previamente e com os planos) por meio dos processos QUA 2 (Verificação) e QUA 3 (Validação).

Uma descrição detalhada sobre as entradas, práticas a serem adotadas, as saídas de cada processo que envolve as atividades de teste pode ser obtido no texto da norma (ISO/IEC 15504-5, 2004).

Como pode ser observado, esta norma define práticas de teste a serem adotadas no processo de desenvolvimento em uma organização de software.

2.3.2.4 – Teste de Software de acordo com o MPS.BR e CMMI

O Projeto MPS.BR – Melhoria de Processo de Software Brasileiro – tem como objetivo fundamental a definição e disseminação de um Modelo de Referência para melhoria de processo de software – MR-MPS (MPS.BR, 2005). Não é objetivo do modelo definir algo novo no que se refere a Normas e Modelos de Maturidade. A novidade e contribuição do projeto estão na sua estratégia de implementação, criada para a realidade brasileira.

O modelo é fortemente baseado nas práticas definidas pelo CMMI (2000), ISO/IEC 12207 (1995) e ISO/IEC 15504 (2003). No entanto, diferentemente do CMMI, este modelo apresenta 7 níveis de maturidade para uma organização de software: (A)

Em Otimização, (B) Gerenciado Quantitativamente, (C) Largamente Definido, (D) Definido, (E) Parcialmente Definido, (F) Gerenciado e (G) Parcialmente Gerenciado. A Tabela 2.1 apresenta as áreas de processo (no MPS.BR são chamados de processos) que compõem os modelos de acordo com cada nível de maturidade.

Tabela 2.1. Estrutura dos modelos MPS.BR e CMMI

Níveis de Maturidade		Áreas de Processo
CMMI	MPS.BR	
5	A	Inovação e Implantação na Organização Análise e Resolução de Causas
4	B	Desempenho do Processo Organizacional
		Gerência Quantitativa do Processo
3	C	Análise de Decisão e Resolução
		Gerência de Risco
	D	Desenvolvimento de Requisitos
		Solução Técnica
		Integração do Produto
		Instalação do Produto
		Liberação do Produto
		Verificação
	E	Validação
		Definição do Processo Organizacional
Avaliação e Melhoria do Processo Organizacional		
Adaptação do Processo para Gerência de Projeto		
2	F	Treinamento
		Gerência de Configuração
		Medição
		Garantia da Qualidade
	G	Aquisição
		Gerência de Projeto
		Gerência de Requisitos

Para atingir um nível de maturidade em ambos os modelos, todas as áreas de processo referentes ao nível desejado e aos níveis inferiores devem ser implementadas. No CMMI, cada área de processo é composta por objetivos, que podem ser genéricos ou específicos, cuja implementação é obrigatória para atingir determinado nível de maturidade. Esses objetivos (genéricos ou específicos) são compostos por práticas, que podem ser genéricas ou específicas, respectivamente.

Em relação às atividades de teste, ambos os modelos abordam essa questão de forma semelhante, com diferenças somente nos níveis de maturidade. As atividades de teste de software não possuem um processo (ou área de processo) específico, ou seja, com esse nome, mas estão fortemente presentes em outros processos que compõem cada modelo. Uma preocupação mais formal com testes de software é descrita somente a partir do Nível D no MPS.BR (Nível 3 CMMI) por meio dos seguintes processos: *Solução Técnica*, *Integração do Produto*, *Verificação* e

Validação, conforme descrito nas Tabelas 2.2, 2.3, 2.4 e 2.5 em textos extraídos do CMMI (2000):

Tabela 2.2. Área de Processo “Solução Técnica”

Área de Processo: Solução Técnica

O Objetivo desta área de processo é projetar, desenvolver e implementar soluções para os requisitos definidos. A seguir está o trecho em que a atividade de teste de software é citada.

Objetivo Específico 3: Implementar o Projeto do Produto

Prática Específica 3.1: Implementar o Projeto

Sub-prática 4: Realizar Testes de Unidades no componente do produto de forma apropriada.

Tabela 2.3. Área de Processo “Integração do Produto”

Área de Processo: Integração do Produto

O Objetivo desta área de processo é integrar o produto a partir de seus componentes, garantindo, desta forma, o seu funcionamento. Como descrito, esta área de processo está relacionada diretamente com a atividade de Teste de Integração.

Objetivo Específico 1: Preparar para Integração do Produto

Objetivo Específico 2: Garantir Compatibilidade de Interfaces

Objetivo Específico 3: Reunir Componentes do produto e Entregar o produto

Tabela 2.4. Área de Processo “Verificação”

Área de Processo: Verificação

Segundo o CMMI, o objetivo desta Área de Processo é garantir que os artefatos estejam de acordo com os seus requisitos especificados, ou seja, que o artefato foi produzido da forma “correta” ou especificada. Segundo esta área de processo, um dos métodos para realizar Verificação é por meio de Testes, e esta área de processo define práticas desde a definição até execução das atividades de verificação.

Objetivo Específico 1: Preparar para Verificação

Prática Específica 1.1: Selecionar produtos e métodos de verificação (entre os quais está teste)

Prática Específica 1.3: Estabelecer Procedimentos e Critérios de Verificação (tipo de teste a ser realizado)

Objetivo Específico 3: Verificar Produto Selecionado (execução dos testes)

Tabela 2.5. Área de Processo “Validação”

Área de Processo: Validação

Segundo o CMMI, o objetivo desta Área de Processo é demonstrar que o produto ou artefato cumpre as funcionalidades para o qual foi intencionado quando localizado no seu ambiente definitivo. Uma das formas de realizar Validação é por meio da realização de tipos específicos de testes. O objetivo desta Área de Processo está de acordo com os objetivos almejados por Teste de Aceitação e Teste Beta (realizado no ambiente do usuário).

Objetivo Específico 1: Preparar para Validação

Prática Específica 1.1: Selecionar produtos e métodos de validação (entre os quais está teste)

Prática Específica 1.3: Estabelecer Procedimentos e Critérios de Validação (tipo de teste a ser realizado)

Objetivo Específico 2: Validar Produto ou Componente (execução dos testes)

Nesta seção pôde ser observado que os diferentes modelos de referência de processo abordam as atividades de teste de forma diferenciada, mas em todos é ressaltada a importância dos testes em um projeto de software, incluindo a

necessidade de um planejamento e controle para os testes. No entanto, estes modelos não descrevem explicitamente algumas particularidades relacionadas ao processo de teste, como a escolha do critério e técnicas de teste, a gerência de defeitos, necessidade de automação dos testes, etc. Com isso, surgiram na literatura modelos de referência específicos para avaliação de processos de testes de software. Na seção seguinte, estes modelos são descritos.

2.3.3 – Modelos de Referência de Processo de Testes de Software

Conforme discutido na seção 2.3.2, os diversos modelos de referência para avaliação de processo de software descrevem as atividades de teste de forma diferenciada, em diferentes níveis de detalhamento e utilizando nomenclatura distinta. Apesar de explicitarem a necessidade de processos durante o desenvolvimento de um produto, inclusive um processo de testes de software, não descrevem algumas particularidades a serem seguidas durante os testes em software.

Com isso, alguns trabalhos (BURNSTEIN e SUWANASSART, 1996; KOOMEN e POL, 1999) surgem com o intuito de caracterizar as necessidades de um processo de testes e avaliá-los com o intuito de prover melhorias neste processo. A seguir, serão destacados dois modelos que contribuem para avaliação e melhorias de um processo de testes: TMM (BURNSTEIN e SUWANASSART, 1996) e TPI (KOOMEN e POL, 1999).

2.3.3.1 – TMM – *Testing Maturity Model*

Os modelos de referência para avaliação da maturidade do processo de desenvolvimento de software, tais como CMM (SEI, 1993) e ISO/IEC 12207 (1995) não abordam particularidades do processo de testes, mas sim trata processo de uma forma genérica e que deve ser planejado e controlado seguindo os mesmos procedimento. Por esta razão, o *Illinois Institute of Technology* criou o TMM – *Test Maturity Model* (BURNSTEIN e SUWANASSART, 1996). O objetivo do TMM, segundo seus idealizadores, é dar suporte às organizações na melhoria do processo de testes e tem as seguintes características:

- É um modelo complementar ao CMM e enfatiza questões relacionadas exclusivamente às atividades de teste de software;
- Provem uma metodologia para avaliar a maturidade do processo de testes de software em uma organização;

- Foi projetado para guiar organizações de software na seleção de estratégias de melhoria no processo de testes, e;
- Provem um método para auxiliar as medições do processo de testes;

Porém, o TMM somente almeja a avaliação de um processo de testes, ou seja, é necessário que exista um processo de testes para a aplicação deste modelo, pois o modelo não apóia a definição de um processo de testes para uma organização. TMM utiliza uma abordagem para avaliação de processos de testes baseada em níveis de maturidade, semelhante ao que é realizado no CMM, e apresenta cinco níveis⁵, que são:

Tabela 2.6. Níveis do modelo TMM

Níveis	Descrição dos níveis
1	Inicial
2	Fase de definição
3	Integração
4	Gestão e medições
5	Otimização, prevenção de defeitos e controle de qualidade

Segundo este modelo, um processo “maduro” possui diversas características:

- Um conjunto de práticas de teste definidas;
- Ciclo de vida dos testes;
- Um processo para planejamento dos testes;
- Um grupo de testes específico;
- Um grupo de melhoria de processo de testes;
- Um conjunto de métricas relacionadas aos testes;
- Ferramentas e equipamentos para as atividades de teste;
- Mecanismo de controle e rastreamento, e;
- Controle da qualidade do produto.

Todos esses pontos citados indicam características relevantes em uma organização que deseja realizar testes de forma organizada, permitindo melhorias constantes no seu processo de testes.

2.3.3.2 – TPI – *Test Process Improvement*

Conforme discutido na seção 2.3.2.1, o CMM, segundo CRAIG e JASKIEL (2002), não aborda adequadamente questões relacionadas às atividades de teste até

⁵ Um detalhamento sobre o modelo e os níveis de maturidade definido em TMM pode ser obtido na referência (BURNSTEIN e SUWANASSART, 1996) citada no Capítulo “Referências Bibliográficas”.

os níveis 1 e 2, que consiste, no presente momento, na situação real de grande parte das organizações. Com isso, Martin Pol e Tim Koomen desenvolveram uma metodologia de melhoria do processo de testes conhecido como TPI – *Test Process Improvement* (KOOMEN e POL, 1999). O TPI fornece um guia para o CMM nível 3 (para testes). Este processo é bem documentado no livro *Test Process Improvement*, onde é apresentado um mecanismo para utilização deste modelo.

Conceitualmente, TPI é um método para caracterização de um processo de testes de software e identificação de oportunidades de melhorias no processo de testes usando um modelo estático (CRAIG e JASKIEL, 2002).

No entanto, de forma semelhante ao descrito no modelo TMM, o modelo TPI somente almeja a avaliação de um processo de testes já existente, e não apóia a definição de um processo de testes em uma organização. Este modelo permite aos seus usuários determinar o estado atual do seu processo de testes através de diversas áreas chaves de processo, semelhante ao que é realizado no CMM, que buscam destacar pontos a serem considerados em um processo de testes através da criação de quatro níveis de maturidade: A, B, C e D (os níveis ascendentes indicam aumento na maturidade do processo. Ex: “B” é mais maduro que “A”).

O objetivo principal do TPI é destacar pontos que devem ser considerados em um processo de testes de uma organização de software. Para isso, apresenta diversas áreas chave de processo, que são:

- Estratégia de Teste;
- Modelo de ciclo de vida;
- Métricas;
- Avaliação dos testes
- Comprometimento e motivação;
- Treinamento de teste;
- Relato dos testes;
- Gerência de defeitos;
- Gerência de *testware*⁶;
- Técnicas de especificação de testes.
- Momento de envolvimento;
- Estimativa e planejamento;
- Automação dos testes;
- Ambientes de teste;
- Escopo de metodologia;
- Comunicação;
- Teste em baixo nível;
- Gerência do processo de testes;

A utilização desses modelos específicos que abordam avaliação de processos de testes pode auxiliar muitas organizações a escolher os passos lógicos seguintes na

⁶ *Testware* se refere aos documentos principais das atividades de teste, em analogia ao que acontece com “software”.

busca de melhorias no seu processo de testes ou a abordagem de teste a ser utilizada em seus projetos.

Além disso, em ambos os modelos é ressaltada a necessidade e importância do planejamento e controle para a melhoria dos testes de software. No entanto, estes modelos não fornecem mecanismos que apoiem a realização de tais atividades, garantindo a qualidade dos testes. Com isso, surge a necessidade de investigar na literatura técnica abordagens que visam ao apoio ao planejamento e controle de testes de software que possam ser aplicadas ao longo do processo de testes em uma organização.

Desta forma, na seção seguinte são apresentadas seis abordagens que apoiam o planejamento e controle dos testes de software identificadas ao longo deste trabalho, caracterizando o estado da arte em relação às atividades de planejamento e controle de testes de software.

2.4 – Utilização de Revisão Sistemática para Identificação de Abordagens para Apoiar o Planejamento e Controle de Testes

Esta seção apresenta a realização de uma pesquisa bibliográfica, utilizando conceitos de revisão sistemática (KITCHENHAM, 2004), com o objetivo de identificar e caracterizar abordagens que apoiem o planejamento e controle de testes em software existente na literatura técnica da área de teste de software.

2.4.1 – Revisão Sistemática

Revisão sistemática é um meio de identificar, avaliar e interpretar toda pesquisa disponível e relevante sobre uma questão de pesquisa, um tópico ou um fenômeno de interesse (KITCHENHAM, 2004). A condução de uma revisão sistemática supostamente apresenta uma avaliação justa do tópico de pesquisa à medida que utiliza uma metodologia de revisão rigorosa, confiável e passível de auditoria (MAFRA e TRAVASSOS, 2005). Além disso, uma revisão sistemática deve obrigatoriamente conter o protocolo de busca executado de forma a permitir que a revisão seja repetida por outros pesquisadores interessados.

Os estudos individuais que são analisados em uma revisão sistemática são chamados de estudos primários. A revisão sistemática em si é considerada um tipo de estudo secundário.

Para este trabalho, foram utilizados os conceitos relacionados à revisão sistemática. No entanto, os passos para realização desta pesquisa não seguiram à

risca o processo para planejamento e execução de revisões sistemáticas apresentado em (KITCHENHAM, 2004). No entanto, foi definido um protocolo para realização da revisão, o que leva à possibilidade de auditoria dos resultados obtidos e repetição do estudo por meio de novas execuções desta revisão.

2.4.2 – Protocolo de Revisão Sistemática para identificação das Abordagens

O protocolo de revisão sistemática seguido foi baseado no modelo disponível em (KITCHENHAM, 2004), e será apresentado a seguir:

- **Objetivo do Estudo:** *Identificação de Abordagens que apoiem a realização de planejamento e controle das atividades de teste de software.*
- **Formulação da pergunta:** *Quais são as abordagens que apoiam o planejamento e controle de testes de software?*
- **Critérios de seleção de fontes:** *disponibilidade de consulta de material por meio da Internet; presença de mecanismos de busca por meio de palavras-chave, descrição em livros técnicos conhecidos pelos pesquisadores.*
- **Métodos de busca de fontes:** *as fontes serão acessadas por meio da Internet ou através de livros técnicos impressos.*
- **Palavras-chave:** *(testing ou test), planning, control, software, (approach ou methodology ou method ou guideline ou process ou systematic).*
- **Listagem de fontes:** *periódicos da CAPES (IEEE journals and conferences, IEE journals, IEE conferences, IEEE conferences, IEEE standards, ACM journals, ACM conferences, Kluwer journals e Elsevier journals), Biblioteca Digital INSPEC, Base de Dados COMPENDEX EI, Livro “Systematic Software Testing” (CRAIG e JASKIEL, 2002).*
- **Tipo dos estudos primários:** *artigos científicos publicados em congressos ou Revistas, padrões internacionais ou abordagem relatadas em livros técnicos.*
- **Critérios de inclusão e exclusão dos estudos primários**
 - *Estudos que apresentarem abordagens para planejamento e controle de testes com um embasamento científico ou que foram caracterizadas experimentalmente;*
 - *Os artigos devem estar disponíveis por meio da Internet;*
 - *Os livros técnicos utilizados devem ser incluídos na descrição da Listagem de Fontes;*

- **Crerios de Qualidade dos estudos primrios:** *Avaliar a origem dos estudos selecionados, dando preferncias para estudos publicados em Revistas, conferncias ou livros tcnicos da rea de Engenharia de Software.*
- **Sumarizao dos resultados:** *os resultados sero tabulados.*

Esta pesquisa foi realizada inicialmente no perodo de 10 de julho a 25 de setembro de 2004 e re-executado em 16 de novembro a 10 de dezembro de 2005. Seguindo estes passos especificados, foram selecionadas seis abordagens para planejamento e controle de testes de software que sero descritas na seo 2.4.3.

2.4.3 – Abordagens para Planejamento e Controle de Testes de Software

Esta seo descreve as seis abordagens que apoiam o planejamento e controle de testes de software que foram identificadas por meio da pesquisa realizada.

2.4.3.1 – IEEE Standard 1059-1993: Guia para Plano de Verificao e Validao (V&V)

Consiste em um guia na preparao de Planos de Verificao e Validao de Software (SVVPs) de acordo com o padro IEEE *Standard 1012-1986* (1986). O IEEE *Std 1012* especifica o contedo necessrio para um SVVP, mas no define passos para o seu preenchimento.

- **IEEE Standard 1012 – 1986: Padro para V&V de Software**

Este padro define o contedo de um SVVP e o processo de V&V em termos de atividades especficas e tarefas relacionadas na sua execuo. Os objetivos deste padro so:

- 1) Estabelecer um *framework* comum para o processo, atividades e tarefas de V&V que apoiam todos os processos de ciclo de vida de software (aquisio, fornecimento, desenvolvimento, operao e manuteno).
- 2) Definir as tarefas de V&V, entradas especificadas e resultados esperados.
- 3) Identificar as tarefas mnimas de V&V correspondente ao nvel de integridade de software, utilizando um esquema de quatro nveis.
- 4) Definir o contedo de um Plano de V&V de Software (SVVP).

Desta forma, este padro apresenta o contedo de um plano de Verificao, no entanto, no guia no seu preenchimento. Este guia aplicado utilizando-se o IEEE *Std 1059* (1993).

- **Guia para planejamento de V&V de software**

O objetivo do planejamento e documentação do plano é empregar os recursos de V&V eficientemente, monitorar e controlar o processo de V&V, permitindo a identificação de cada papel e responsabilidade dos participantes de um projeto. Planejamento resulta em um SVVP bem estruturado e realístico que fornece fundamentos para o sucesso na aplicação de V&V. Documentação do escopo e declaração explícita dos objetivos facilitam o entendimento de todas as partes envolvidas sobre o que está sendo feito para que os objetivos sejam atingidos.

O planejamento de V&V, que deve ser uma parte integrada do planejamento global do projeto, pode ser dividido em alguns passos:

- Identificar o escopo de V&V;
- Estabelecer objetivos específicos a partir do escopo geral;
- Analisar as entradas do projeto antes de selecionar ferramentas, técnicas e preparar plano;
- Selecionar técnicas e ferramentas de teste, e;
- Desenvolver o plano de testes.

Desta forma, este guia define passos a serem seguidos na realização do planejamento de V&V utilizando o IEEE *Std* 1012 (1986) como forma de documentação.

2.4.3.2 – IEEE *Standard* 1008-1987: Padrão para Teste de Unidade

O IEEE *Standard* 1008 (1987) define uma abordagem integrada para sistematizar e documentar testes de unidade. A abordagem utiliza informações de projeto e implementação de unidades, além de seus requisitos, para determinar a completude dos testes.

O escopo deste padrão é limitado à atividade de Teste de Unidade (checagem de pequenas partes, módulos do sistema).

O objetivo deste padrão é definir uma abordagem para testes de unidade que pode ser utilizada como base para várias práticas de Engenharia de Software e fornecer um guia para auxiliar na implementação e utilização desta abordagem.

O padrão define três passos a serem realizados:

- Planejamento dos testes;
- Definição do conjunto de casos e procedimentos de teste, e;
- Medição dos testes de unidade.

Em relação à atividade de planejamento, são definidos os recursos, abordagens utilizadas, cronogramas, características a serem testadas, valores de entradas e resultados esperados de acordo com o plano de teste geral para o projeto (definido, segundo este padrão, por meio do IEEE *Std* 829 (1983)).

A aquisição do conjunto de teste consiste em projetar os testes, definindo os casos de teste. E a medição dos testes consiste na execução dos casos de teste, concluir a execução e avaliar as saídas geradas.

2.4.3.3 – *Rapid Test Planning*

Abordagem para Planejamento de testes desenvolvida por KANER e BACH (2001) que tem como objetivo principal simplificar o planejamento, gerenciamento e documentação das atividades de teste de software.

Segundo KANER e BACH (2001), a melhor abordagem para testes depende exclusivamente do contexto do projeto. Há o caso em que é necessária uma documentação detalhada, em outros casos não há essa necessidade. Então, antes da implementação de alguma diretriz, os requisitos do sistema devem ser analisados.

A necessidade de uma documentação específica deve ser avaliada antes de se tomarem decisões sobre métodos, padrões ou tecnologias. Devem ser considerados os seguintes aspectos:

- Pessoas envolvidas no projeto (*stakeholders*);
- Interesses dos envolvidos em relação ao documento;
- Ações (o que será feito com a documentação);
- Objetos (quais documentos possuem alto ou baixo valor).

A partir das necessidades de cada projeto, é proposto um modelo para testes de software (Figura 2.3) composto pelos seguintes elementos que devem ser considerados no planejamento das atividades de teste:

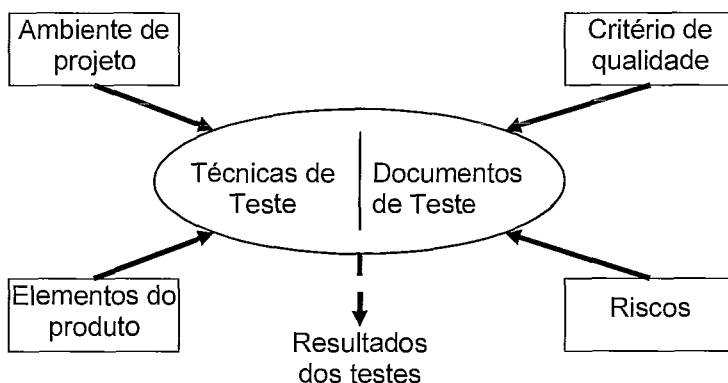


Figura 2.3 - Elementos da abordagem *Rapid Test Planning*

2.4.3.4 – STEP – Systematic Test and Evaluation Process

A metodologia STEP – *Systematic Test and Evaluation Process* – foi desenvolvida por HETZEL (1985) como uma forma de implementar o padrão para documentação de teste original IEEE *Std* 829 (1983) e foi atualizado baseando-se na última versão do IEEE *Std* 829 em 1998 (1998) e no IEEE *Std* 1008 (1987) para Testes de Unidade de Software.

STEP define um modelo de processo para testes de software. A metodologia é composta por: tarefas específicas, produtos gerados e papéis definidos (Figura 2.4), e não é dependente de ferramentas e não assume qualquer organização ou funcionários específicos.

Papéis:	Tarefas:	Produtos gerados:
Gerente	Planejar estratégia	Documentação
Analista	Adquirir Testware	Procedimentos
Técnico		Dados
Revisor	Medir comportamento	Software de apoio

Figura 2.4 - Elementos de STEP (CRAIG e JASKIEL, 2002)

As tarefas são: Planejar a estratégia (definição dos planos de teste), Adquirir o *testware*, Medir o comportamento (executar os testes, checar os resultados e avaliar o processo). A Figura 2.5 descreve a temporização dessas atividades de acordo com os níveis de teste. Os produtos gerados são dos documentos definidos na norma IEEE *Std* 829 (1998).

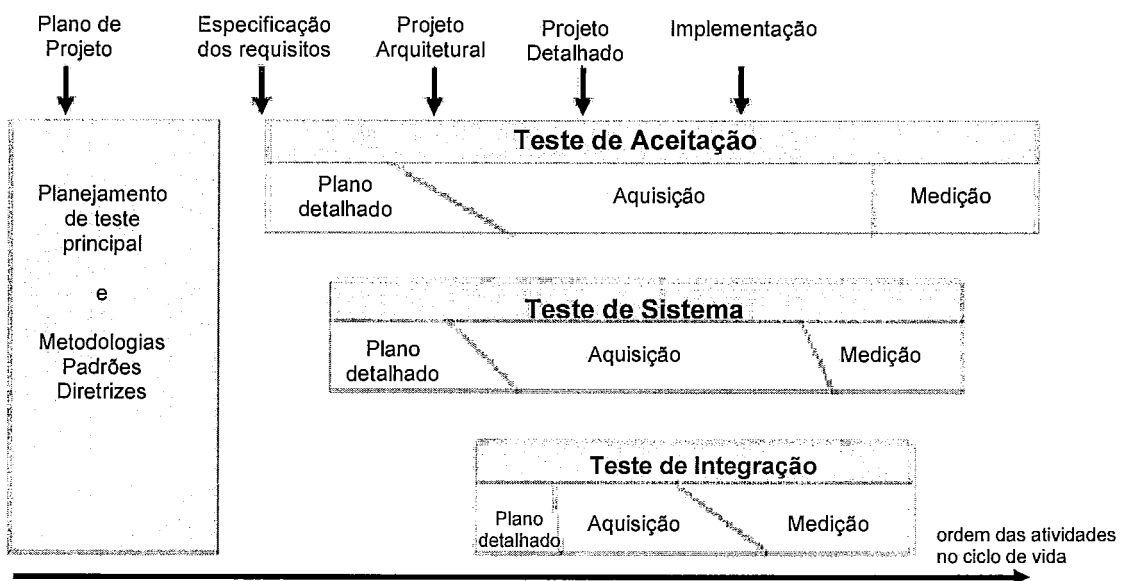


Figura 2.5 – Temporização das atividades de acordo com o nível de Teste (CRAIG e JASKIEL, 2002)

Os papéis e suas responsabilidades especificados na metodologia STEP estão apresentados na Tabela 2.7

Tabela 2.7 - Papéis e responsabilidades em STEP

Papéis	Descrição das responsabilidades
Gerente	Comunicar, planejar e coordenar
Analista	Planejar, estimar, projetar e avaliar
Técnico	Implementar, executar e checar os dados
Revisor	Examinar e avaliar

Portanto, STEP é uma metodologia orientada à prevenção e enfatiza os defeitos de requisitos e projeto de sistema. Suas atividades iniciam durante a definição dos requisitos de um software e seguem até o final do desenvolvimento. Outro aspecto importante de STEP é a criação de grupos de casos de teste com cobertura conhecida (mapeando casos de testes entre requisitos, projetos e códigos).

2.4.3.5 – ABORDAGEM PARA PLANEJAMENTO DE TESTE A PARTIR DE REQUISITOS DO SISTEMA – TEST PLANNER

Em SAMSON (1993) é apresentado um *framework* para sistema baseado em conhecimento que prover assistência para a preparação de plano de teste de sistema a partir dos requisitos do sistema.

O objetivo da abordagem apresentada consiste na especificação de um planejador de testes baseado em conhecimento (*Test Planner*) que identifique testes de sistemas em alto nível a partir dos seus requisitos.

O escopo da abordagem apresentada se restringe à atividade de Teste de aceitação, utilizando os requisitos de um sistema como mecanismos para o planejamento dos testes.

Segundo SAMSOM (1993), a integração das atividades de planejamento dos testes com as atividades de definição de requisitos facilita a rastreabilidade entre características do sistema e requisitos. Rastreabilidade também apóia uma nova validação do plano de teste quando mudanças são feitas nos requisitos durante o desenvolvimento.

A abordagem apresentada é composta por três componentes principais: taxonomia de requisitos, taxonomia de tipos de teste e processo de combinação entre requisitos e tipos de teste (Figura 2.6).

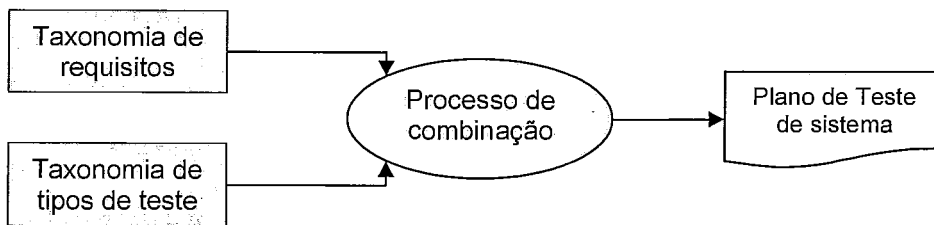


Figura 2.6 – Arquitetura conceitual de *Test Planner*

A taxonomia de requisitos foi desenvolvida para facilitar a classificação dos requisitos e seu relacionamento dentro do *framework*. A utilização de uma taxonomia foi utilizada por três fatores: reduzir a linguagem evitando os riscos de erros de interpretação, para fazer aproximação do processamento de um texto em linguagem natural para classificação dos requisitos e com a abordagem de classificação, testadores não precisam aprender uma nova linguagem. A taxonomia de requisitos foi obtida a partir de vários estudos experimentais.

A taxonomia de tipos de teste funciona de forma semelhante à taxonomia de requisitos e permite combinar os tipos de requisitos com os tipos de teste.

As atividades descritas para o *framework* são: classificar requisitos a partir de uma descrição textual de acordo com a taxonomia e combinar requisitos com tipos de teste utilizando conhecimento prévio, feito por meio de algoritmos de buscas.

2.4.3.6 – BATOOM: UMA ABORDAGEM PRÁTICA PARA TESTES DE SOFTWARE ORIENTADO A OBJETOS

Em (DENG e HE, 1998) é apresentada uma abordagem para planejamento, execução e documentação de testes de software orientado a objetos. Nesse trabalho são descritas algumas diferenças entre testes de software orientado a objetos e testes de software específico.

Essa abordagem sugere a divisão da equipe de um projeto entre engenheiros de desenvolvimento (equipe de desenvolvedores) e engenheiros de qualidade (equipe de teste). Além disso, prevê um conjunto de atividades a serem realizadas pelos engenheiros de qualidade durante os testes. Essas atividades são:

- Planejamento e configuração dos objetivos dos testes de software e requisitos de teste;
- Refinamento e especificação dos planos de teste detalhados e esboço dos testes, construindo casos, procedimentos e cenários de teste;
- Execução e re-execução dos casos de teste (manuais e automatizados);

- Relato dos testes, indicando os resultados obtidos, os problemas detectados e uma análise dos dados.

BATOOM pode ser aplicada em três níveis de teste: Teste de Unidade, Teste de Integração e Teste de Sistema. Nessa abordagem é descrito um mecanismo para sincronização entre as atividades de teste e de depuração, conforme descrito na Figura 2.7. Dessa forma, os testes são divididos em três fases e possuem dois pontos de controle.

As fases são:

1. **Testes primitivos:** o foco dos testes está nas principais funcionalidades e operações comuns do software que está sendo testado;
2. **Testes detalhados:** após a correção dos defeitos identificados nos testes primitivos, são realizados testes em cada área funcional do software. Testes de funcionalidades, performances, interface, usabilidade, compatibilidade, instalação, stress, quando necessários, são realizados.
3. **Testes de regressão:** o objetivo é avaliar que todas as funcionalidades estão funcionando como previsto e que todos os problemas detectados tenham sido resolvidos, garantindo que novos defeitos não foram introduzidos entre versões do produto.

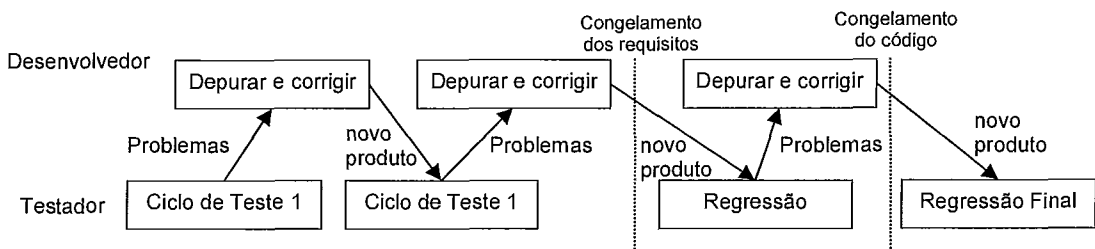


Figura 2.7 – O mecanismo para paralelismo entre desenvolvimento e testes

Os pontos de controle são:

- **Congelamento dos requisitos:** momento em que todas os requisitos funcionais e não-funcionais foram implementados. Não ocorrerá mudança nos requisitos do software.
- **Congelamento do código:** momento em que todos os testes de regressão, com exceção do teste final de regressão onde todo o sistema será executado novamente, estão completados. Todos os problemas detectados foram resolvidos, e não ocorrerão mudanças de código, a não ser em caso de futuras manutenções.

BATOOM possui, ainda, um mecanismo para documentação dos testes de acordo com cada etapa que está sendo realizada. Os documentos são: Plano do

Produto, Plano de Teste Mestre, Esboço dos Testes, Relatório de Problema do Software, Relatório de Execução dos Testes e Relatório de Completude dos Testes.

Apesar da descrição dessa abordagem, não foram identificados artigos científicos que relatassem os modelos dos documentos utilizados por BATOOM e nem resultados da sua aplicação em ambientes reais ou acadêmicos avaliando a sua eficiência ou comparando-a com outras abordagens existentes.

2.4.4 – Caracterização das Abordagens Identificadas

Os critérios de caracterização foram definidos baseando-se em dois aspectos:

- **Características específicas das atividades de planejamento e controle, segundo o PMBOK (2004):** documentação gerada, atividades realizadas, identificação de riscos;
- **Características específicas e relevantes às atividades de teste de software:** escopo dos testes, critério de geração de casos de teste⁷ e facilidades de rastreabilidade entre requisitos e casos de teste.

2.4.4.1 – Características Específicas das Atividades de Planejamento e Controle

- **Documentação:** avaliar o tipo de informação gerada após o planejamento e controle dos testes e analisa a utilização de algum padrão de documentação. O PMBOK (2004) descreve a necessidade de documentação para facilitar a comunicação entre equipes. A documentação deve estar consistente e coerente com o restante do projeto.
- **Atividades realizadas para o planejamento:** avaliar os passos a serem seguidos durante o planejamento dos testes. O PMBOK (2004) descreve a necessidade de especificar quais atividades devem ser realizadas e a seqüência de execução.
- **Identificação dos riscos:** O PMBOK (2004) define o processo de risco como um processo facilitador da atividade de planejamento e ressalta a sua importância.

2.4.4.2 – Características Específicas e relevantes às atividades de Teste de Software

- **Escopo:** analisar o tipo de teste apoiado pela abordagem (unidade, integração, sistema, aceitação ou regressão). Isso deverá ser utilizado como base para o restante do planejamento e controle dos testes de software.

⁷ Ver definição de Critério de Geração de Casos de Teste na Seção 2.1.

- **Critério de geração de casos de teste:** avaliar a forma na qual são gerados os casos de teste, levando-se em consideração o critério de cobertura dos testes⁸. Segundo MYERS (1979), um dos vários fatores que influenciam no custo das atividades de teste consiste no número de casos de teste projetado, visto que a cada teste, recursos (físicos e humanos) devem ser alocados.
- **Facilidade de rastreabilidade entre requisitos e casos de testes:** avaliar a existência de mecanismos que facilite a realização de testes de regressão após modificações nos requisitos do software. MATS (2001) discute a falta de rastreabilidade entre requisitos e casos de testes como um dos principais problemas em teste de software.

Na Tabela 2.8 é apresentado o resultado da caracterização das abordagens de planejamento e controle de testes de acordo com os critérios definidos.

Os dados apresentados refletem somente uma caracterização analítica das abordagens identificadas de acordo com alguns pontos identificados. Esta comparação não permite a obtenção de evidências sobre a aplicabilidade de cada abordagem identificada em relação ao seu comportamento e desempenho referente ao planejamento e controle dos testes durante um processo de desenvolvimento real.

Durante a realização deste trabalho, foram pesquisadas referências sobre a avaliação experimental de abordagens para planejamento, mas não foram obtidos estudos que tratassem este aspecto. Somente avaliações sobre técnicas específicas de geração e seleção de casos de teste ou análise da eficiência dessas técnicas foram identificadas, como por exemplo em (JURISTO et al., 2005), que descreve o conjunto de estudos experimentais identificado na literatura técnica e que estão relacionados às técnicas de teste realizados ao longo de 25 anos de pesquisa em teste de software.

⁸ Ver definição de Critério de Cobertura dos Testes na Seção 2.1.

2.4.4.3 – Resultados da Caracterização das Abordagens de Planejamento e Controle de Teste de Software

Os resultados da caracterização das abordagens estão apresentados na Tabela 2.8. Os dados utilizados na tabela foram coletados a partir das referências disponibilizadas por cada abordagem identificada.

Tabela 2.8 - Caracterização das abordagens de planejamento e controle de teste de software

Abordagem / Critério	Documentação	Atividades realizadas	Identificação de Riscos	Escopo dos testes	Geração de casos de teste	Facilidade de rastreabilidade
IEEE Std 1059	Segue o IEEE Std 1012	1. Identificar o escopo de V&V 2. Estabelecer objetivos específicos 3. Analisar as entradas do projeto 4. Selecionar técnicas e ferramentas 5. Desenvolver o plano	O	Todos os níveis de teste	Escolhida a partir do projeto	O
IEEE Std 1008	Segue o IEEE Std 829	1. planejamento dos testes 2. aquisição do conjunto de teste 3. medição dos testes de unidade	O	Teste de Unidade	Escolhida a partir do projeto	O
<i>Rapid Test Planning</i>	Documentação própria (tabelas e matrizes)	Especificação de: • Ambiente de projeto • Critérios de qualidade • Elementos do produto • Riscos • Técnicas de testes • Documentos gerados	O	Todos os níveis de teste	Escolhida a partir do projeto	X
STEP	Segue o IEEE Std 829	1. Planejar estratégia 2. Adquirir <i>Testware</i> 3. Medir comportamento	O	Todos os níveis de teste	Escolhida a partir do projeto	O
<i>Test Planner</i>	Não definido	1. classificar requisitos a 2. combinar requisitos e tipos de teste	X	Teste de aceitação	Geração a partir de requisitos	O
BATOOM	Documentação própria • Plano do Produto • Plano de Teste Mestre • Esboço dos Testes • Relatório de Problema do Software • Relatório de Execução dos Testes • Relatório de Completude dos Testes	1. Planejar e configurar objetivos dos testes e requisitos de teste; 2. Refinar e especificar plano de teste detalhado e esboço dos testes; 3. Executar e re-executar casos de teste; 4. Relatar resultados dos testes;	X	Teste de Unidade Teste de Integração Teste de Sistema	Escolhida a partir do projeto	O

Legenda para a Tabela 2.8:

O – Indica que o critério é aplicado entre as atividades da abordagem avaliada.

X – Indica que o critério não é aplicado entre as atividades da abordagem avaliada.

2.5 – Conclusão

Ao longo deste Capítulo, foi realizada uma caracterização do estado da arte sobre as atividades de planejamento e controle dos testes por meio das características do processo de testes de software e de abordagens identificadas que apóiam a realização dessas atividades.

Observou-se a importância das atividades de planejamento e controle para a realização de qualquer atividade. Essa preocupação estende-se à medida que muitos recursos são alocados, custos são introduzidos, riscos são identificados e pessoas são envolvidas.

O planejamento e controle dos testes de software são fatores determinantes para o sucesso desta atividade. No entanto, a ausência de estudos científicos que abordam esse tema pode, eventualmente, dificultar a escolha de estratégias para a realização do planejamento e controle dos testes em uma organização.

Algumas abordagens que visam ao apoio das atividades de planejamento e controle de teste de software estão presentes na literatura, cada uma com características, particularidades e limitações específicas. No entanto, observa-se a ausência de estudos experimentais que avaliem a aplicação de tais abordagens em ambientes reais de desenvolvimento de software, o que dificulta a obtenção de um conhecimento amplo sobre a aplicabilidade e efetividade dessas abordagens naquilo que elas se propõem.

A partir de resultados tão dispersos obtidos a partir deste levantamento bibliográfico, há uma grande dificuldade em entender como as atividades de teste de software estão sendo aplicadas na prática, ou seja, como isto está sendo aplicado em projetos de software reais desenvolvidos por organizações de software. Portanto, torna-se necessário entender como isto vem ocorrendo neste ambiente específico, para que a partir dessas informações possam ser identificadas eventuais necessidades, ou carências, dessas organizações e conseqüentemente oportunidades de melhorias.

Com isso, o Capítulo seguinte, "*Pesquisa de Opinião para Caracterização de Práticas de Teste em Organizações de Software*", tem como objetivo principal descrever uma caracterização inicial do estado da prática da área de teste de software no Brasil em um cenário de desenvolvimento de software específico por meio da realização de uma pesquisa de opinião. Com isso, deseja-se observar como algumas práticas de teste de software, sugeridas pela literatura técnica, estão sendo aplicadas em ambientes reais de desenvolvimento de software no contexto onde a pesquisa foi conduzida.

Capítulo 3

Pesquisa de Opinião para Caracterização de Práticas de Teste em Organizações de Software

Neste capítulo são apresentados o planejamento, execução e resultados de uma pesquisa de opinião realizada em organizações de software, localizadas em um cenário específico de desenvolvimento de software, com o objetivo de obter uma caracterização inicial de como práticas de teste de software vêm sendo aplicadas na indústria, e como os profissionais que atuam nas organizações julgam a importância dessas práticas.

3.1 – Introdução

Pesquisa de Opinião⁹ é uma forma de investigação realizada, normalmente, por meio de entrevistas ou questionários com o objetivo de obter uma opinião de pessoas sobre um determinado assunto ou contexto (WOHLIN *et al.*, 2000). A partir dos dados coletados, os resultados são analisados para obtenção de conclusões descritivas ou exploratórias.

No presente trabalho, uma pesquisa de opinião foi aplicada com o propósito de investigar o estado da prática sobre Verificação e Validação (V&V), particularmente, as atividades de revisão e teste de software, em organizações de software brasileiras. As questões foram extraídas da literatura relacionadas à revisão e teste de software e buscam capturar informações sobre a aplicabilidade e o grau de importância que as práticas de revisão e teste de software possuem nas organizações.

Espera-se que os resultados indiquem, para o cenário do estudo, quais e de que forma as práticas relacionadas à revisão e teste de software estão sendo utilizadas na indústria de software.

⁹ Survey.

Conforme descrito anteriormente, este estudo visa caracterizar as atividades de revisão e teste de software, mas apesar do planejamento e execução desta pesquisa envolver essas duas atividades, somente os dados relacionados a teste de software fazem parte do escopo deste trabalho e somente estes serão analisados.

Pesquisadores e engenheiros de software podem se beneficiar desta pesquisa de opinião por pelo menos três formas diferentes. Primeiro, a pesquisa pode ser utilizada para capturar e determinar o estado da prática de organizações específicas, sob o propósito de caracterização. Estas organizações podem identificar seus pontos de melhoria após conhecer seu estado de maturidade nestas áreas abordadas pela pesquisa. O segundo benefício é servir aos pesquisadores como um ponto de partida a futuras pesquisas para a melhoria do apoio às práticas de revisão e teste de software. E terceiro, pesquisadores da engenharia de software experimental podem se beneficiar das lições aprendidas para se construir e executar futuras pesquisas de opinião na área de engenharia de software.

Dois estudos piloto foram realizados em um ambiente acadêmico nos dias 09 e 16 de maio de 2005, cada estudo com dois participantes com o objetivo de avaliar os dados dos questionários e as dificuldades na sua aplicação. Baseado nos resultados destes estudos piloto, os questionários foram evoluídos para facilitar a aplicação do estudo real. No entanto, os dados coletados durante a execução dos dois estudos pilotos não foram analisados sob o propósito de caracterização das atividades de revisão e teste de software.

Sendo assim, este Capítulo está organizado da seguinte forma: na seção seguinte são apresentadas as etapas de Definição e Planejamento da pesquisa, seguindo o processo para condução de um estudo experimental descrito em MENDONÇA (2005). Na Seção 3.3 são apresentadas a etapa de Execução do estudo e os resultados quantitativos obtidos. Na Seção 3.4 é descrita a análise qualitativa dos resultados obtidos no estudo. Por fim, na Seção 3.5 são descritas as conclusões obtidas durante este capítulo.

3.2 – Definição e Planejamento do Estudo

Nesta seção são descritos os objetivos do estudo, o contexto de aplicação, as hipóteses definidas, as variáveis selecionadas e uma descrição da instrumentação a ser utilizada durante a sua execução.

3.2.1 – Objetivo do Estudo

O objetivo do estudo é analisar um conjunto de práticas de verificação (revisão) e validação (teste) de software, extraídas da literatura, com relação à sua aplicação e importância em organizações de desenvolvimento de software sob o ponto de vista de gerentes e desenvolvedores de software.

A Tabela 3.1 representa o objetivo do estudo seguindo o modelo definido pelo método GQM (BASILI *et al.*, 1994).

Tabela 3.1. Objetivos do Estudo seguindo o método GQM

<p>OBJETIVO 1: <i>Analisar um conjunto de práticas relacionadas à Revisão e Teste de Software</i> Com o propósito de caracterizar Com respeito à aplicabilidade Do ponto de vista de profissionais da área de desenvolvimento de software No contexto de projetos de organizações de desenvolvimento de software</p> <p>OBJETIVO 2: <i>Analisar um conjunto de práticas relacionadas à Revisão e Teste de Software</i> Com o propósito de caracterizar Com respeito ao grau de importância Do ponto de vista de profissionais da área de desenvolvimento de software No contexto de organizações de desenvolvimento de software</p>

Assim, pretende-se avaliar as seguintes questões gerais:

- Q1: Quais práticas avaliadas são realizadas pelas organizações?
- Q2: Quais práticas avaliadas são julgadas importantes?

3.2.2 – Definição das Hipóteses

Para cada objetivo do estudo, foi elaborada uma hipótese nula e uma hipótese alternativa para o plano do estudo:

Para avaliação da aplicabilidade:

- **Hipótese Nula 1 (H_0 1):** as organizações não aplicam práticas de revisão e teste de software, segundo os participantes do estudo.

H_0 1: $(A_{RevA} + A_{TesA}) = \emptyset$ e $(A_{RevN} + A_{TesN}) = A_T$, onde:

- A_T = Lista total de práticas relativas à Revisão e Teste de software avaliadas quanto à sua aplicabilidade.

- A_{RevN} = Lista de práticas relativas à revisão de software que não são aplicadas pelas organizações segundo os participantes do estudo.
- A_{RevA} = Lista de práticas relativas à revisão de software que são aplicadas pelas organizações segundo os participantes do estudo.
- A_{TesN} = Lista de práticas relativas à teste de software que não são aplicadas pelas organizações segundo os participantes do estudo.
- A_{TesA} = Lista de práticas relativas à teste de software que são aplicadas pelas organizações segundo os participantes do estudo.

$$A_T = (A_{RevN} + A_{RevA}) + (A_{TesN} + A_{TesA})$$

- **Hipótese Alternativa 1 (H_1 1):** as organizações utilizam ao menos uma das práticas de revisão e teste de software, segundo os participantes do estudo.

$$H_1 \text{ 1: } (A_{RevA} + A_{TesA}) \neq \emptyset \text{ e } (A_{RevN} + A_{TesN}) \neq A_T$$

Para avaliação do grau de importância:

- **Hipótese Nula 2 (H_0 2):** todas as práticas de revisão e teste de software são considerados “não importantes”, segundo os participantes do estudo.

$$H_0 \text{ 2: } (G_{RevI} + G_{TesI}) = \emptyset \text{ e } (G_{RevN} + G_{TesN}) = G_T, \text{ onde:}$$

- G_T = Lista total de práticas relativas à revisão e teste de software avaliadas quanto ao seu grau de importância.
- G_{RevN} = Lista de práticas relativas à revisão de software que não são consideradas importantes pelos participantes do estudo.
- G_{RevI} = Lista de práticas relativas à revisão de software que são consideradas importantes pelos participantes do estudo.
- G_{TesN} = Lista de práticas relativas à teste de software que não são consideradas importantes pelos participantes do estudo.
- G_{TesI} = Lista de práticas relativas à teste de software que são consideradas importantes pelos participantes do estudo.

$$G_T = (G_{RevN} + G_{RevI}) + (G_{TesN} + G_{TesI})$$

- **Hipótese Alternativa 2 (H_1 2):** há ao menos uma prática de revisão e teste de software considerada importante, segundo os participantes do estudo.

$$H_1 \text{ 2: } (G_{RevI} + G_{TesI}) \neq \emptyset \text{ e } (G_{RevN} + G_{TesN}) \neq G_T$$

3.2.3 – Seleção das Variáveis

Este estudo utiliza as seguintes variáveis:

- **Variáveis independentes:**
 - O conjunto de práticas relacionadas à revisão de software extraído da literatura.
 - O conjunto de práticas relacionadas à teste de software extraído da literatura.

- **Variáveis dependentes:**
 - O conjunto de práticas relacionadas à revisão e teste de software que não são aplicadas pelas organizações ($A_{RevN} + A_{TesN}$).
 - O conjunto de práticas relacionadas à revisão e teste de software que são aplicadas pelas organizações ($A_{RevA} + A_{TesA}$).
 - O conjunto de práticas relacionadas à revisão e teste de software que não são consideradas importantes, sob o ponto de vista dos participantes ($G_{RevN} + G_{TesN}$).
 - O conjunto de práticas relacionadas à revisão e teste de software que são consideradas importantes, sob o ponto de vista dos participantes ($G_{RevI} + G_{TesI}$).

3.2.4 – Descrição da Instrumentação

Neste estudo, foi desenvolvido um questionário a ser utilizado para coleta dos dados. O questionário contém cinco seções:

1. Formulário para caracterização da organização participante do estudo, avaliando seu perfil, tamanho, tipo de software desenvolvido e certificações.
2. Formulário para caracterização do participante do estudo, avaliando sua formação e experiência nas atividades de revisão e teste de software.
3. Formulário para avaliação de práticas de revisão de software.
4. Formulário para avaliação de práticas de teste de software.
5. Glossário com os termos técnicos citados, para facilitar o entendimento dos participantes.

Os questionários do estudo foram elaborados a partir de informações descritas em trabalhos disponíveis na literatura que descrevem e avaliam práticas de revisão e teste:

- (GELPERIN e HETZEL, 1988): Este trabalho apresenta os resultados de uma pesquisa realizada em um congresso internacional sobre teste de software realizada em 1987 nos EUA onde avaliou-se 20 práticas de teste de software a respeito da sua aplicação na indústria;
- (LAITENBERGER *et al.*, 2002): Este trabalho apresenta um estudo (e seus resultados) onde avaliou-se o estado da prática de revisão e inspeção de software em organizações da Alemanha no ano de 2002.
- (ANDERSSON e RUNESON, 2002): Este trabalho apresenta uma pesquisa de opinião que avaliou o estado da prática de verificação e validação de software na indústria realizada em 2002.
- (CRAIG e JASKIEL, 2002): Livro com o título "Teste de Software Sistemático", onde são discutidos vários assuntos relacionados a teste de software, e ao final, apresenta uma pesquisa de opinião que avalia 66 práticas de teste de software.
- (SEPIN, 2005): Relatório que apresenta indicadores sobre a qualidade e produtividade do setor de software brasileiro publicado pelo Ministério de Ciência e Tecnologia. Os dados deste trabalho foi utilizado para definição do formulário de caracterização das organizações participantes do estudo;
- Além desses trabalho, utilizou-se a experiência científica e prática do Grupo de Engenharia de Software da COPPE/UFRJ nas áreas de verificação e validação de software.

Para avaliação do nível de aplicabilidade de uma prática de teste de software em uma organização, foram definidas escalas ordinais, mais precisamente uma escala de Likert (JAMIELSON, 2004), que variam de 0 a 4, com o objetivo de obter respostas mais precisas sobre a sua aplicabilidade nas organizações. As respostas expressam um aumento gradativo no nível de aplicação de uma prática nos projetos da organização:

0. Não Aplicável – não faz parte do contexto da organização.
1. Não Utilizado pela organização.
2. Uso Não Freqüente: Usado poucas vezes.
3. Uso Habitual: Usado em grande parte dos projetos de software da organização.
4. Uso Padrão: Usado em todos os projetos de software da organização.

Para avaliação do grau de importância de uma prática de teste de software, também se adotou uma escala ordinal composta por 5 opções, pelos mesmos motivos da

escala de aplicabilidade, e elas expressam um aumento gradativo no grau de importância de uma prática na opinião de cada profissional da organização:

0. Não Importante: Não necessário.
1. Valor Baixo: Pouco importante.
2. Valor Limitado: Poderia ser adequadamente utilizado.
3. Valor Significante: Prática recomendada.
4. Valor Essencial: Deve ser uma prática padrão para todos.

Ao total, são 32 itens, que correspondem a práticas de teste de software, avaliados no questionário. Os itens foram projetados de forma a facilitar uma análise qualitativa dos resultados por meio de uma verificação hierárquica das respostas obtidas. Os itens do questionário se dividem, hierarquicamente, nos seguintes grupos, conforme apresentado na Figura 3.1:

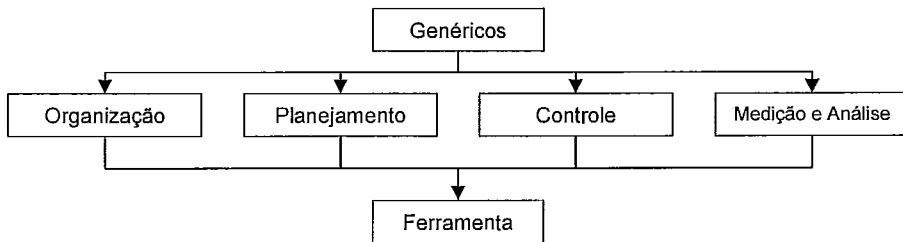


Figura 3.1. Hierarquização entre os itens do questionário.

- **Genéricos:** itens do topo da hierarquia que indicam simplesmente o nome de tipos de teste a serem realizados, e são:
 - Realização de Teste de Unidade.
 - Realização de Teste de Integração.
 - Realização de Teste de Sistema.
 - Realização de Teste de Aceitação.
 - Realização de Teste de Regressão.
- **Organização:** itens relacionados à organização das atividades de teste de software por parte da direção da empresa. Eles referem-se a decisões tomadas pelos responsáveis pelos projetos ou pela organização, e são:
 - Existência de um responsável ou uma equipe alocada para as atividades de teste.
 - Realização de treinamentos sobre testes.
 - Separação das atividades de teste do desenvolvimento.
 - Existência de testadores em tempo integral para realização de testes.

- Utilização de metodologia ou processo para sistematizar e organizar testes de software.
- Re-execução dos testes quando o software é modificado.
- **Planejamento:** itens que estão relacionados exclusivamente a atividades de planejamento dos testes, e são:
 - Documentação do Plano de Teste descrevendo objetivos/abordagens.
 - Documentação de Procedimentos e Casos de Teste e seus Resultados.
 - Identificação e utilização de riscos para projetar, organizar e executar testes de software.
 - Desenvolvimento de testes de software antes da codificação.
- **Controle:** itens que estão relacionados exclusivamente ao controle dos testes de software em um projeto ou em uma organização, e são:
 - Registro do tempo gasto em testes.
 - Registro de falhas e defeitos detectados durante os testes.
 - Conservação de registro (*log*) dos testes executados.
 - Conservação dos dados dos testes para usos futuros.
 - Monitoração da aderência ao processo de teste de software.
 - Avaliação dos documentos de teste (*testware*) quanto à sua qualidade.
- **Medição e Análise:** itens relacionados às atividades de medição e análise dos testes, como mecanismo para obter uma maior interpretação dos dados obtidos durante os testes e criar uma base histórica de projetos, possibilitando uma melhoria constante nas atividades de software em uma organização, e são:
 - Medição dos testes e rastreamento dos custos com os testes.
 - Medição da densidade de defeito (defeitos por mil linhas de código).
 - Análise regular do padrão (tendência) de falhas e defeitos encontrados.
 - Análise e medição da cobertura dos testes.
 - Análise de defeitos identificados durante os testes para descobrir suas causas e como foram introduzidos (rastreabilidade).
- **Ferramenta:** itens que estão no terceiro nível da hierarquia dos itens (Figura 3.1) e estão relacionados à utilização de ferramentas que apoiem a realização de alguma das tarefas que compõem o nível 2 da hierarquia dos itens (Organização, Planejamento, Controle e Medição e Análise), e são:

- Disponibilidade de um banco de dados de teste (pacote de testes que simulam o ambiente de teste) para reutilização.
- Utilização de ferramentas geradoras de procedimentos/casos de testes.
- Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste.
- Utilização de ferramentas para estimar esforço de teste ou cronograma.
- Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.
- Utilização de ferramentas de medida de cobertura nas fases de teste (unidade, integração, sistema).

Os questionários de caracterização de organização e indivíduos e de avaliação de práticas de V&V elaborados como forma de instrumentação estão apresentados no Apêndice A deste trabalho.

3.2.5 – Contexto de Aplicação do Estudo

De acordo com (WOHLIN, 2000), o contexto pode ser caracterizado conforme quatro dimensões:

- Processo: *on-line* / *off-line*;
- Participantes: alunos / profissionais;
- Realidade: problema real / modelado;
- Generalidade: específico / geral.

As dimensões deste estudo são: **off-line** / **profissionais** / **problema real** / **específico**. Eis a explicação a seguir:

Este estudo se propõe a ser executado de forma **off-line**, uma vez que não haverá um monitoramento contínuo sobre as atividades dos participantes. Não há restrições no número de pessoas em cada organização que pode participar do estudo. Cada participante terá o seu tempo e ambiente para preencher o questionário, colaborando com o estudo. Os participantes serão **profissionais** com experiência na área de desenvolvimento de software e preferencialmente com participação anterior em atividades de verificação e validação. O estudo é **específico**, pois é realizado para as áreas de Revisão e Teste de Software e endereça um **problema real**, isto é, a caracterização dessas práticas realizadas cotidianamente em organizações de software.

Para cada organização de software visitada, devem ser realizadas três etapas para a execução do estudo.

- (1) Inicialmente, deve ser entregue um questionário de caracterização da organização à pessoa responsável pela organização. Para cada organização, somente um questionário de caracterização de organização deve ser preenchido.
- (2) Após isso, devem ser entregues os questionários de consentimento dos participantes do estudo, onde é realizada a caracterização de cada participante do estudo.
- (3) Após este preenchimento, devem ser entregues os questionários de avaliação das práticas de revisão e teste de software e o glossário de termos aos participantes do estudo (gerentes de projeto e desenvolvedores).

3.3 – Execução e Resultados do Estudo

Esta seção descreve os dados coletados durante a aplicação do estudo e seus resultados quantitativos.

3.3.1 – Cenário de Aplicação do Estudo

O estudo foi conduzido em organizações de software localizadas no pólo de desenvolvimento de software do Estado do Amazonas. Este cenário foi selecionado por conveniência, facilitado pelo incentivo oferecido pela Agência de Fomento a Pesquisa do Estado do Amazonas (FAPEAM¹⁰), aliado ao interesse das organizações deste pólo em participar desta pesquisa.

3.3.2 – Dados Obtidos

O estudo foi conduzido no período de 28/06/2005 a 13/07/2005 por meio de visitas às empresas contatadas e que se interessaram em participar do estudo. Os questionários foram preenchidos sempre na própria empresa em que os participantes do estudo atuavam e com o consentimento da alta direção.

Apesar de não ser necessário, conforme descrito no planejamento do estudo, o executor do estudo acompanhou o preenchimento dos questionários para evitar que os

¹⁰ Fundação de Amparo à Pesquisa do Estado do Amazonas – www.fapeam.am.gov.br.

participantes deixassem de responder alguma questão por causa de eventuais más interpretações. Sendo assim, quando solicitado, o executor do estudo esclareceu as dúvidas dos participantes durante o preenchimento do questionário. Ao final deste período, foram obtidos os seguintes dados:

- **Organizações de software visitadas:** 13 organizações.

As organizações que participaram do estudo desenvolvem diferentes categorias de software e possuem diferentes tamanhos, conforme descrito no gráfico apresentado na Figura 3.2. O critério utilizado para definição das categorias de organizações a partir de seu tamanho foi definido pelo Ministério de Ciência e Tecnologia (SEPIN, 2005) e está apresentado na Tabela 3.2.

Tabela 3.2 – Critério para definição das categorias de organizações.

Categoria	Tamanho da Organização
Micro empresa	Menos que 10 funcionários
Pequenas empresas	10 a 49 funcionários
Médias empresas	50 a 99 funcionários
Grandes empresas	Mais que 100 funcionários

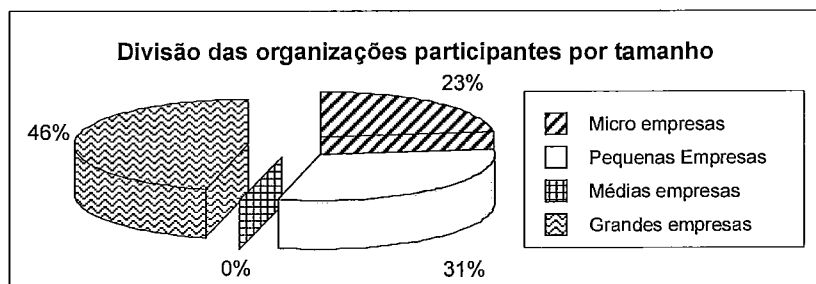


Figura 3.2. Divisão das organizações que participaram do estudo por tamanho

- **Questionários Respondidos:** 36 questionários (distribuídos entre as 13 organizações visitadas).

Além disso, os profissionais que participaram do estudo possuem diferentes níveis de formação e experiência nas atividades de revisão e teste de software, conforme descrito no gráfico apresentado na Figura 3.3.

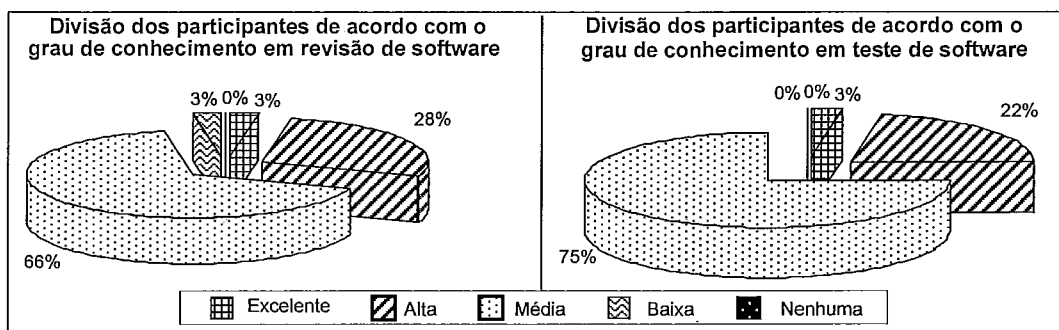


Figura 3.3. Divisão dos profissionais que participaram do estudo por experiência

Os dados completos com as caracterizações das organizações, participantes e as respostas de todos os participantes do estudo estão disponíveis no Apêndice B deste trabalho.

3.3.3 – Resultados do Estudo

As seções seguintes descrevem os procedimentos adotados para análise dos dados relacionados aos dois objetivos da pesquisa: **aplicabilidade** e **grau de importância** de cada item questionado.

3.3.3.1 – Resultados da avaliação da Aplicabilidade de Práticas de Teste de Software nas Organizações

A avaliação da aplicabilidade indica como as práticas de teste que compõem o questionário do estudo estão sendo utilizadas nos projetos de software das organizações. Com isso, para cada prática de teste de software avaliada, torna-se necessário obter um resultado único para cada organização visitada, pois uma organização pode ter mais que um participante no estudo. O objetivo é analisar a distribuição das respostas obtidas em cada organização entre as escalas possíveis do estudo (as escalas variam de 0 a 4, conforme descrito na seção 3.2.4). Para isso, foram definidos alguns passos para avaliação da aplicabilidade das práticas de teste:

- **PASSO 1:** Para cada item do questionário, obteve-se a moda (o resultado mais freqüente) entre os valores obtidos, chegando a um consenso dentro da empresa, conforme exemplificado na Tabela 3.3. Este procedimento foi adotado pois se utilizou

uma escala de Likert (JAMIELSON, 2004), que sugere a utilização da moda como mecanismo de análise dos dados de uma amostra.

Tabela 3.3 – Exemplo da obtenção da moda para cada item do questionário.

Organização: Exemplo – Item do questionário: Item X	RESPOSTA
PARTICIPANTE A	3
PARTICIPANTE B	2
PARTICIPANTE C	3
MODA OBTIDA	3

PASSO 2: Em seguida, os dados de todas as organizações visitadas foram agrupados (Tabela 3.4) e distribuídos entre as escalas do estudo para cada item do questionário (Tabela 3.5).

Tabela 3.4 – Exemplo de agrupamento das respostas das organizações.

Item do questionário: Item X	RESPOSTA
ORGANIZAÇÃO A	1
ORGANIZAÇÃO B	3
ORGANIZAÇÃO C	1
ORGANIZAÇÃO D	2
ORGANIZAÇÃO E	4
ORGANIZAÇÃO F	2

Tabela 3.5 – Exemplo da distribuição dos dados entre as escala para o *Item X* de acordo com a Tabela 3.4.

Escala (Item X)	Total de Organizações
0 – Não aplicável	0
1 – Não utilizado	2
2 – Uso Não Freqüente	2
3 – Uso Habitual	1
4 – Uso Padrão	1

- **Avaliação Global das Organizações**

A partir da execução desses passos com os dados obtidos na execução do estudo, avaliaram-se os resultados de todas as organizações simultaneamente, ou seja, sem a divisão da amostra em grupos. Com isso, chegou-se aos resultados apresentados na Tabela 3.6 e visualizado na Figura 3.4, por meio de um gráfico de barras (indicado para apresentação da distribuição das observações do estudo em uma escala de Likert):

Tabela 3.6 – Resultados de Aplicabilidade obtidos no estudo.

TOTALIZACAO DOS RESULTADOS DE APLICABILIDADE						
ITENS QUESTIONADOS		QUANTIDADE DE EMPRESAS POR ESCALA				
		E0	E1	E2	E3	E4
A1	Existência de um responsável ou uma equipe alocada para as atividades de teste.	0	3	0	4	6
A2	Realização de Teste de Unidade.	0	2	2	3	6
A3	Realização de Teste de Integração.	0	0	2	4	7
A4	Realização de Teste de Sistema.	0	0	1	4	8
A5	Realização de Teste de Aceitação.	1	1	4	3	4
A6	Realização de Teste de Regressão.	2	2	4	2	3
A7	Registro do tempo gasto em testes.	2	3	4	2	2
A8	Medição dos testes e rastreamento dos custos com os testes.	2	5	5	1	0
A9	Registro de falhas e defeitos detectados durante os testes.	0	1	0	6	6
A10	Conservação de registro (log) dos testes executados.	0	3	1	4	5
A11	Medição da densidade de defeito (defeitos por mil linhas de código).	1	8	2	1	1
A12	Realização de treinamentos sobre testes.	0	5	5	1	2
A13	Separação das atividades de teste do desenvolvimento.	0	3	2	5	3
A14	Conservação dos dados dos testes para usos futuros.	0	3	2	4	4
A15	Análise regular do padrão (tendência) de falhas e defeitos encontrados.	1	4	4	3	1
A16	Existência de testadores em tempo integral para realização de testes.	1	4	2	1	5
P1	Documentação do Plano de Teste descrevendo objetivos/abordagens.	0	1	1	4	7
P2	Documentação de Procedimentos e Casos de Teste e seus Resultados.	0	1	2	3	7
P3	Registro de falhas e defeitos detectados durante os testes.	0	1	0	5	7
P4	Análise e medição da cobertura dos testes.	0	6	2	2	3
P5	Utilização de metodologia ou processo para sistematizar e organizar testes de software.	0	2	2	4	5
P6	Análise de defeitos identificados durante os testes para descobrir suas causas e como foram introduzidos (rastreamento).	0	4	3	3	3
P7	Identificação e utilização de riscos para projetar, organizar e executar testes de software.	1	4	6	2	0
P8	Desenvolvimento de testes de software antes da codificação.	0	6	5	2	0
P9	Monitoração da aderência ao processo de teste de software.	0	6	3	2	2
P10	Re-execução dos testes quando o software é modificado.		1	5	1	6
P11	Avaliação dos documentos de teste (testware) quanto à sua qualidade.	1	3	2	5	2
F1	Disponibilidade de um banco de dados de teste (pacote de testes que simulam o ambiente de teste) para reutilização.	0	5	5	0	3
	Utilização de ferramentas geradoras de procedimentos ou casos de testes.	1	6	5	0	1
F3	Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste.	0	7	3	1	2
F4	Utilização de ferramentas para estimar esforço de teste e/ou cronograma.	0	8	3	1	1
F5	Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.	0	7	3	1	2
F6	Utilização de ferramentas de medida de cobertura nas fases de teste (unidade, integração, sistema).	0	-	-	0	3

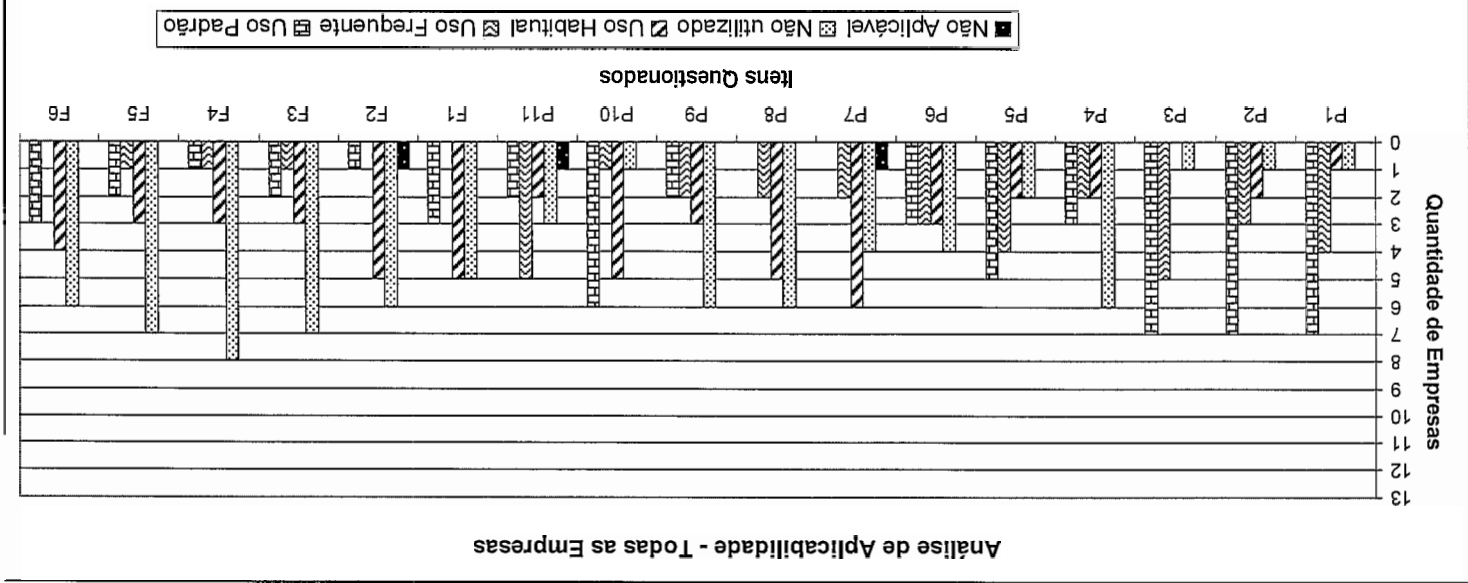
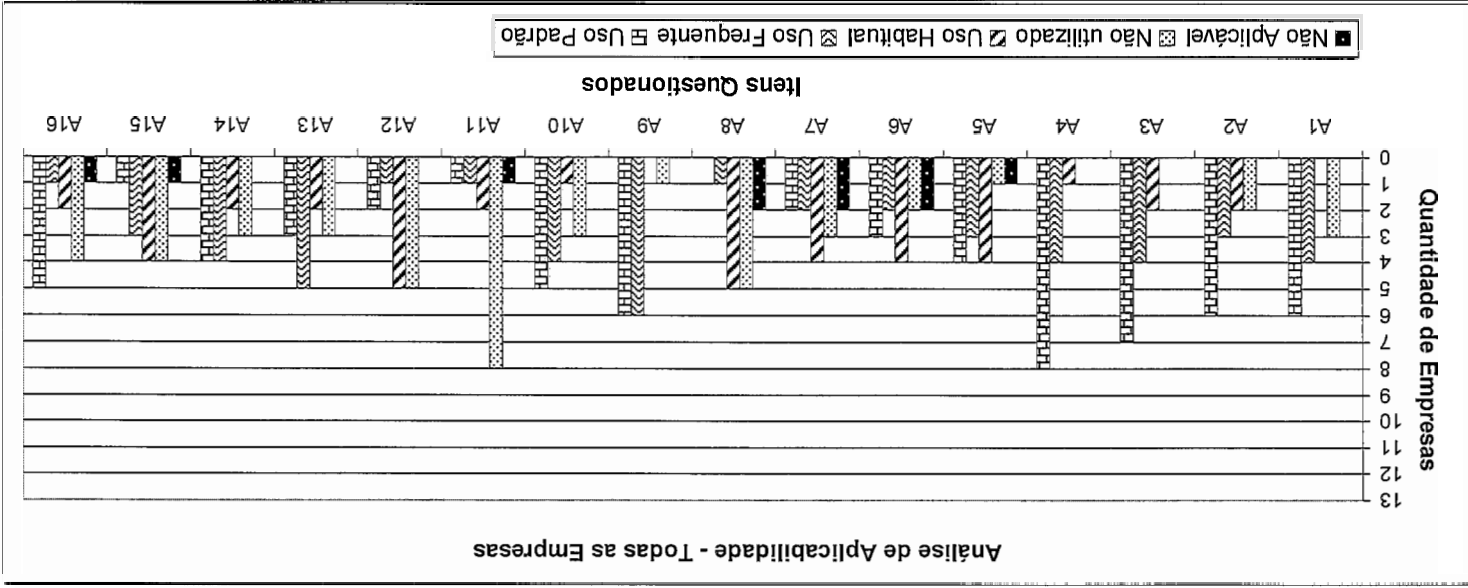


Figura 3.4. Visualização gráfica dos resultados de aplicabilidade obtidos no estudo

- **Avaliação de acordo com o tamanho das Organizações**

No entanto, com a avaliação global dos dados obtidos no estudo, isto é, sem separá-los em grupos, percebeu-se que os resultados de determinados itens que foram questionados eram bastante dispersos, ou seja, existiam organizações que sempre aplicavam um item do questionário e outras que nunca aplicavam o mesmo item, como pode ser observado na Tabela 3.6.

Com isso, optou-se pela divisão da amostra em grupos de acordo com o tamanho de cada organização (micro, pequena e grande empresa) como uma forma de reduzir o viés causado por grandes empresas, que tendem a aplicar mais práticas de Engenharia de Software em seus projetos, pois eles são normalmente mais complexos que os projetos de micros e pequenas empresas. Dessa forma, a amostra foi dividida da seguinte forma:

- 7 empresas ficaram no grupo de micro e pequenas empresas;
- 6 empresas ficaram no grupo de grandes empresas;

Os resultados desse agrupamento estão apresentados nas Tabelas 3.7 e 3.8 e visualizados graficamente nas Figuras 3.5 e 3.6.

Tabela 3.7 – Resultados de Aplicabilidade em Micros e Pequenas Empresas.

RESULTADOS DE APLICABILIDADE – MICROS E PEQUENAS EMPRESAS						
ITENS QUESTIONADOS		QUANTIDADE DE EMPRESAS POR ESCALA				
		E0	E1	E2	E3	E4
A1	Existência de um responsável ou uma equipe alocada para as atividades de teste.	0	3	0	2	2
A2	Realização de Teste de Unidade.	0	0	1	2	4
A3	Realização de Teste de Integração.	0	0	1	2	4
A4	Realização de Teste de Sistema.	0	0	1	2	4
A5	Realização de Teste de Aceitação.	0	1	2	3	1
A6	Realização de Teste de Regressão.	1	2	2	1	1
A7	Registro do tempo gasto em testes.	1	3	1	2	0
A8	Medição dos testes e rastreamento dos custos com os testes.	1	3	3	0	0
A9	Registro de falhas e defeitos detectados durante os testes.	0	1	0	4	2
A10	Conservação de registro (log) dos testes executados.	0	3	0	3	1
A11	Medição da densidade de defeito (defeitos por mil linhas de código).	0	6	0	1	0
A12	Realização de treinamentos sobre testes.	0	5	1	1	0
A13	Separação das atividades de teste do desenvolvimento.	0	3	1	2	1
A14	Conservação dos dados dos testes para usos futuros.	0	3	1	2	1
A15	Análise regular do padrão (tendência) de falhas e defeitos encontrados.	1	3	2	1	0
A16	Existência de testadores em tempo integral para realização de testes.	1	4	0	0	2
P1	Documentação do Plano de Teste descrevendo objetivos/abordagens.	0	1	1	2	3
P2	Documentação de Procedimentos e Casos de Teste e seus Resultados.	0	1	1	2	3
P3	Registro de falhas e defeitos detectados durante os testes.	0	1	0	3	3
P4	Análise e medição da cobertura dos testes.	0	5	1	0	1
P5	Utilização de metodologia ou processo para sistematizar e organizar testes de software.	0	2	1	2	2
P6	Análise de defeitos identificados durante os testes para descobrir suas causas e como foram introduzidos (rastreadibilidade).	0	3	1	2	1
P7	Identificação e utilização de riscos para projetar, organizar e executar testes de software.	0	2	4	1	0
P8	Desenvolvimento de testes de software antes da codificação.	0	3	3	1	0
P9	Monitoração da aderência ao processo de teste de software.	0	4	2	0	1
P10	Re-execução dos testes quando o software é modificado.	0	1	3	0	3
P11	Avaliação dos documentos de teste (testware) quanto à sua qualidade.	1	2	2	2	0
F1	Disponibilidade de um banco de dados de teste (pacote de testes que simulam o ambiente de teste) para reutilização.	0	4	2	0	1
F2	Utilização de ferramentas geradoras de procedimentos ou casos de testes.	0	4	3	0	0
F3	Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste.	0	5	1	1	0
F4	Utilização de ferramentas para estimar esforço de teste e/ou cronograma.	0	6	1	0	0
F5	Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.	0	5	2	0	0
F6	Utilização de ferramentas de medida de cobertura nas fases de teste (unidade, integração, sistema).	0	4	2	0	1

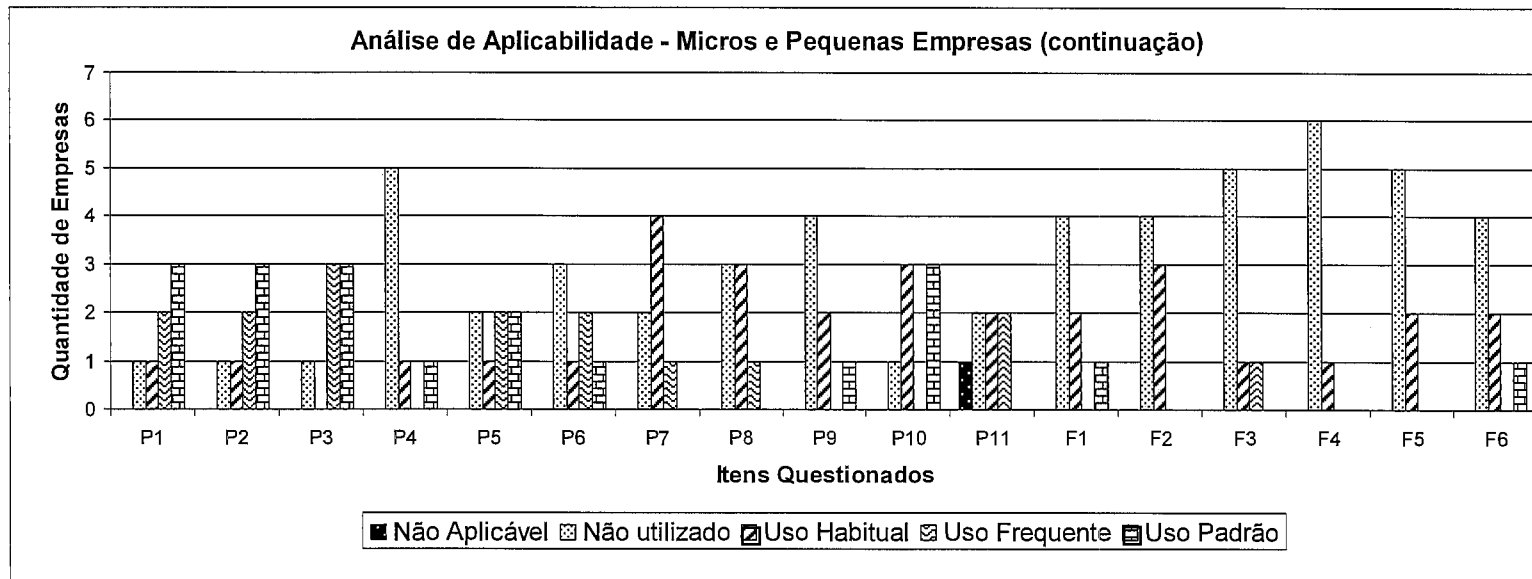
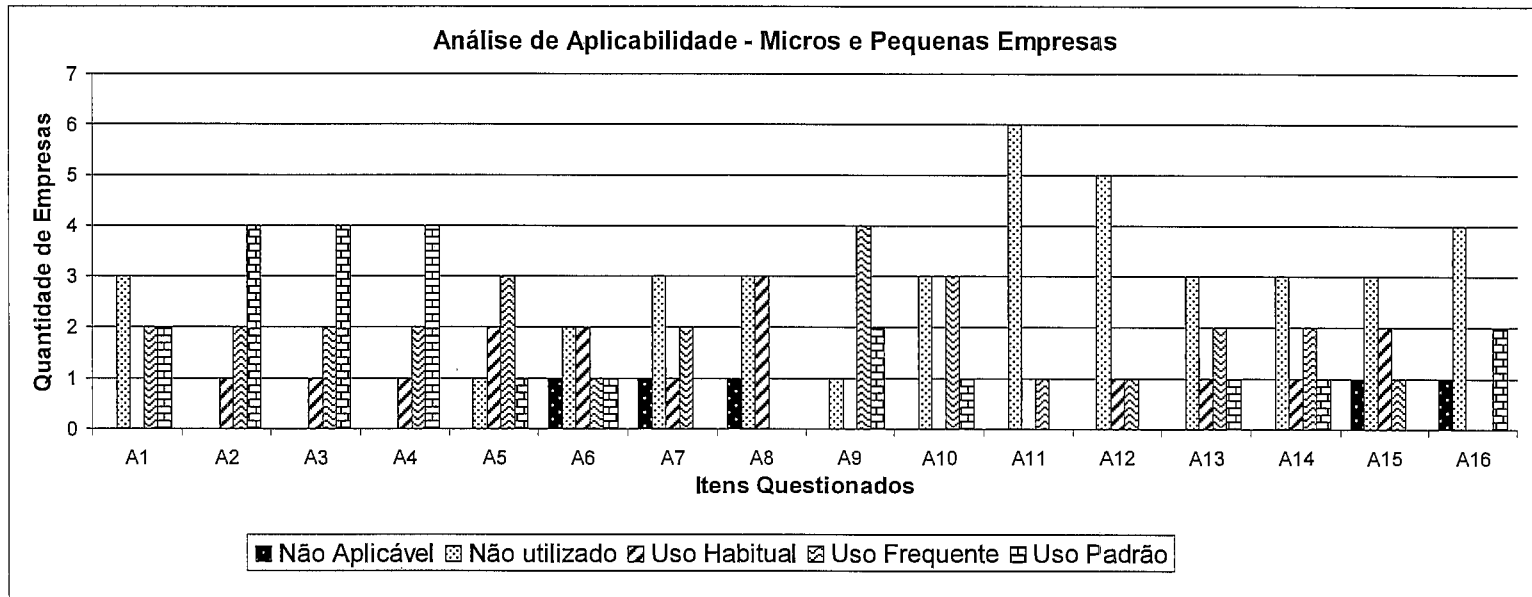


Figura 3.5. Visualização gráfica dos resultados de aplicabilidade obtidos no estudo em Micros e Pequenas Empresas

Tabela 3.8 – Resultados de Aplicabilidade em Grandes Empresas obtidos no estudo.

RESULTADOS DE APLICABILIDADE – GRANDES EMPRESAS						
ITENS QUESTIONADOS		QUANTIDADE DE EMPRESAS POR ESCALA				
		E0	E1	E2	E3	E4
A1	Existência de um responsável ou uma equipe alocada para as atividades de teste.	0	0	0	2	4
A2	Realização de Teste de Unidade.	0	2	1	1	2
A3	Realização de Teste de Integração.	0	0	1	2	3
A4	Realização de Teste de Sistema.	0	0	0	2	4
A5	Realização de Teste de Aceitação.	1	0	2	0	3
A6	Realização de Teste de Regressão.	1	0	2	1	2
A7	Registro do tempo gasto em testes.	1	0	3	0	2
A8	Medição dos testes e rastreamento dos custos com os testes.	1	2	2	1	0
A9	Registro de falhas e defeitos detectados durante os testes.	0	0	0	2	4
A10	Conservação de registro (log) dos testes executados.	0	0	1	1	4
A11	Medição da densidade de defeito (defeitos por mil linhas de código).	1	2	2	0	1
A12	Realização de treinamentos sobre testes.	0	0	4	0	2
A13	Separação das atividades de teste do desenvolvimento.	0	0	1	3	2
A14	Conservação dos dados dos testes para usos futuros.	0	0	1	2	3
A15	Análise regular do padrão (tendência) de falhas e defeitos encontrados	0	1	2	2	1
A16	Existência de testadores em tempo integral para realização de testes.	0	0	2	1	3
P1	Documentação do Plano de Teste descrevendo objetivos/abordagens.	0	0	0	2	4
P2	Documentação de Procedimentos e Casos de Teste e seus Resultados.	0	0	1	1	4
P3	Registro de falhas e defeitos detectados durante os testes.	0	0	0	2	4
P4	Análise e medição da cobertura dos testes.	0	1	1	2	2
P5	Utilização de metodologia ou processo para sistematizar e organizar testes de software.	0	0	1	2	3
P6	Análise de defeitos identificados durante os testes para descobrir suas causas e como foram introduzidos (rastreamento).	0	1	2	1	2
P7	Identificação e utilização de riscos para projetar, organizar e executar testes de software.	1	2	2	1	0
P8	Desenvolvimento de testes de software antes da codificação.	0	3	2	1	0
P9	Monitoração da aderência ao processo de teste de software.	0	2	1	2	1
P10	Re-execução dos testes quando o software é modificado.	0	0	2	1	3
P11	Avaliação dos documentos de teste quanto à sua qualidade.	0	1	0	3	2
F1	Disponibilidade de um banco de dados de teste (pacote de testes que simulam o ambiente de teste) para reutilização.	0	1	3	0	2
F2	Utilização de ferramentas geradoras de procedimentos ou casos de testes.	1	2	2	0	1
F3	Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste.	0	2	2	0	2
F4	Utilização de ferramentas para estimar esforço de teste e/ou cronograma.	0	2	2	1	1
F5	Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.	0	2	1	1	2
F6	Utilização de ferramentas de medida de cobertura nas fases de teste (unidade, integração, sistema).	0	2	2	0	2

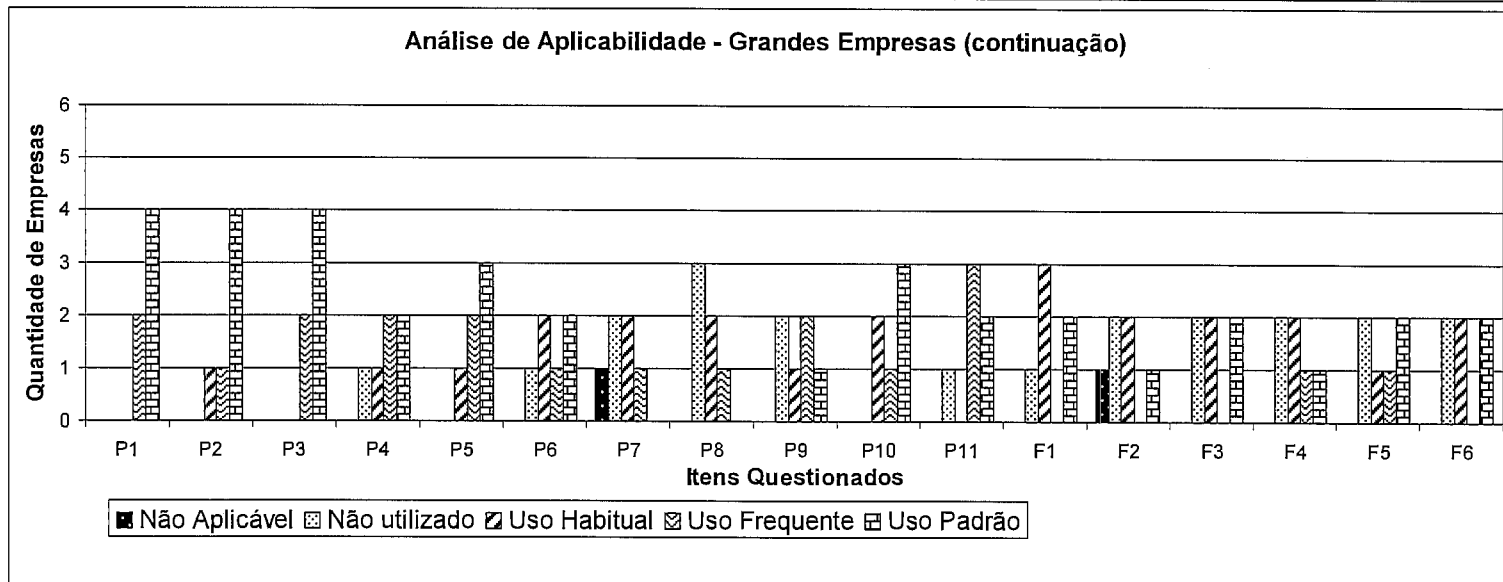
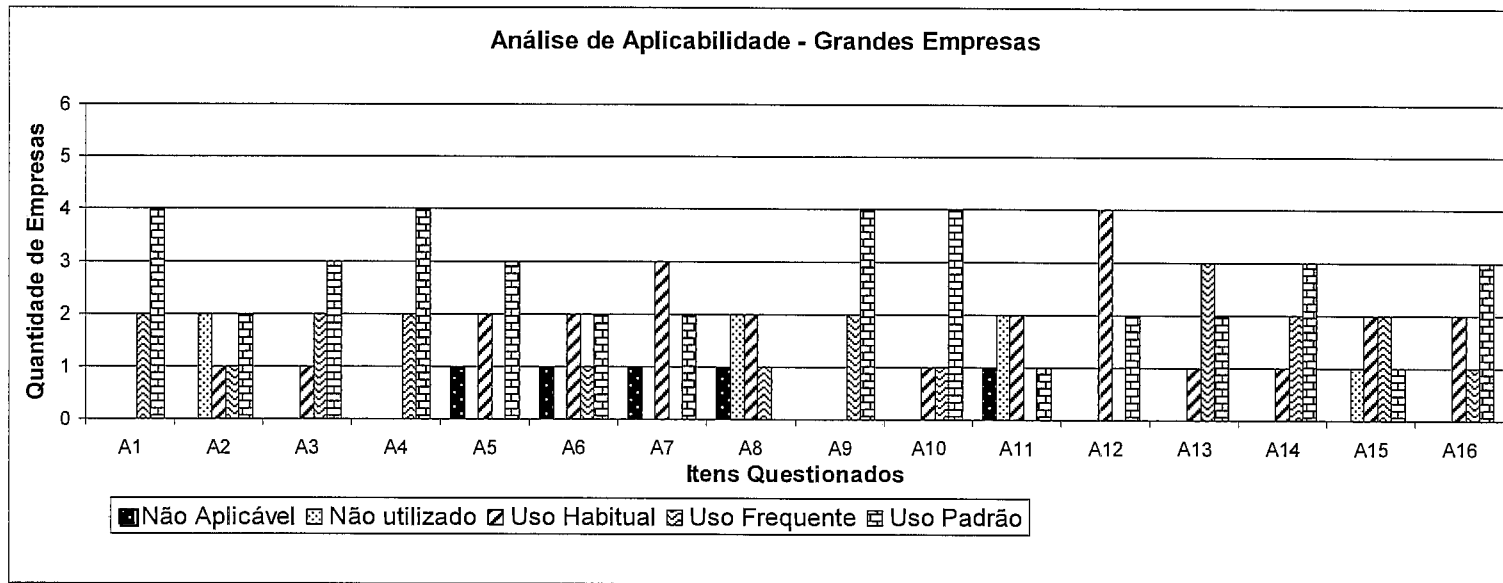


Figura 3.6. Visualização gráfica dos resultados de aplicabilidade obtidos no estudo em Grandes Empresas

3.3.3.2 – Resultados da avaliação do Grau de Importância de Práticas de Teste de Software segundo a opinião dos profissionais das organizações

A avaliação do grau de importância indica a opinião exclusiva das pessoas que responderam aos questionários do estudo, de acordo com a sua formação e seu grau de experiência. Com isso, para cada prática de teste de software avaliada, toma-se necessário obter o seu grau de aplicabilidade considerando o peso de cada participante dentro da amostra obtida a partir do seu nível de formação e grau de experiência. Para isso, foram definidos alguns passos para avaliação do grau de importância das práticas:

- **PASSO 1:** Para cada participante do estudo foi obtido um peso.

O peso dado a cada participante considera quatro perspectivas: a formação acadêmica, o número de projetos em que o participante atuou, o tempo de experiência com desenvolvimento de software e o grau de conhecimento em verificação e validação de software. A fórmula utilizada para definir o peso atribuído aos participantes foi baseada na proposta de FARIAS (2002) e COSTA (2005) para caracterização de gerentes de projetos de software, e adaptada a este estudo. A fórmula é:

$$PT(i) = \frac{TA(i)}{MedianaTA} + \frac{QP(i)}{MedianaQP} + f(i) + g(i) + h(i), \text{ onde:}$$

- $PT(i)$ é o peso total de um participante i ;
- $TA(i)$ é o tempo de atuação na área de desenvolvimento de software de um participante i ;
- $MedianaTA$ é a mediana¹¹ do tempo de atuação, considerando o tempo de atuação de cada participante do estudo;
- $QP(i)$ é a quantidade de projetos que um participante i já esteve envolvido;
- $MedianaQP$ é a mediana da quantidade de projetos que um participante i já esteve envolvido, considerando o número de projetos de cada participante do estudo;
- $f(i)$ é a formação acadêmica na área de Ciência da Computação, sendo considerados os seguintes valores:

$$f(i) = 0, \text{ se o participante possuir nível técnico;}$$

¹¹ Utilizou-se a mediana, e não a média, para evitar que valores extremos distorcessem o valor representativo do Tempo de Atuação e da Quantidade de Projetos.

$f(i) = 1$, se o participante possuir nível superior;

$f(i) = 2$, se o participante possuir especialização;

$f(i) = 3$, se o participante possuir mestrado;

$f(i) = 4$, se o participante possuir doutorado;

$g(i)$ é o conhecimento do participante em relação à verificação de software, sendo considerados os seguintes valores:

$g(i) = 0$, se for nenhum;

$g(i) = 1$, se for baixo;

$g(i) = 2$, se for médio;

$g(i) = 3$, se for alto;

$g(i) = 4$, se for excelente;

$h(i)$ é o conhecimento do participante em relação à validação de software, sendo considerados os seguintes valores:

$h(i) = 0$, se for nenhum;

$h(i) = 1$, se for baixo;

$h(i) = 2$, se for médio;

$h(i) = 3$, se for alto;

$h(i) = 4$, se for excelente;

- **PASSO 2:** Em seguida, multiplicou-se a resposta de cada participante pelo seu peso e calculou-se o valor total para cada item, conforme exemplificado na Tabela 3.9.

Tabela 3.9 – Exemplo da obtenção da moda para cada item do questionário.

Item: Item X	RESPOSTA	PESO
PARTICIPANTE A	2	1,0
PARTICIPANTE B	4	3,0
PARTICIPANTE C	1	0,5
TOTAL	$(2 \times 1,0 + 4 \times 3,0 + 1 \times 0,5) = 14,5$	4,5

- **PASSO 3:** Por fim, normalizou-se o valor obtido no passo 2 para cada prática de teste, ou seja, dividiu-se o valor obtido pelo valor máximo possível: RESPOSTA TOTAL / (PESO TOTAL x 4[escala máxima]), obtendo-se, assim, o grau de importância de um item, que é um valor expresso em percentual (entre 0% e 100%).

$$14,5 / (4,5 \times 4) = 0,80 = 80\%$$

De forma semelhante ao que foi feito na avaliação da aplicabilidade (seção 3.3.2.1), a amostra foi analisada de forma global, ou seja, todas os participantes agrupados em um único grupo, e em categoria de organizações de acordo com o seu tamanho (micros e pequenas empresas ou grandes empresas). O objetivo desta divisão era reduzir o viés ocasionado pelos participantes que atuam em grandes organizações e normalmente conhecem e aplicam atividades de Engenharia de Software com mais frequência em suas tarefas, e conseqüentemente possam vir a distorcer os resultados, dando mais importância a determinados itens que fazem parte do dia-a-dia de suas tarefas.

Com isso, foram obtidos 2 grupos:

- **Participantes que atuam em micro ou pequenas empresas:** 14 participantes;
- **Participantes que atuam em grandes empresas:** 22 participantes;

A partir desses da execução desses passos para os dados obtidos na execução do estudo, chegou-se aos resultados apresentados na Tabela 3.10.

Tabela 3.10 – Resultados de Grau de Importância obtidos no estudo.

ITENS QUE COMPUSERAM O QUESTIONÁRIO NO SURVEY		MICRO E PEQUENA	GRANDE	GERAL
A1	Existência de um responsável ou uma equipe alocada para as atividades de teste.	89,91%	100,00%	95,98%
A2	Realização de Teste de Unidade.	90,80%	92,82%	92,02%
A3	Realização de Teste de Integração.	90,80%	92,02%	91,54%
A4	Realização de Teste de Sistema.	89,31%	92,08%	90,98%
A5	Realização de Teste de Aceitação.	92,34%	97,63%	95,53%
A6	Realização de Teste de Regressão.	85,70%	83,43%	84,33%
A7	Registro do tempo gasto em testes.	73,18%	89,41%	82,95%
A8	Medição dos testes e rastreamento dos custos com os testes.	68,43%	80,40%	75,63%
A9	Registro de falhas e defeitos detectados durante os testes.	87,01%	93,53%	90,94%
A10	Conservação de registro (log) dos testes executados.	70,59%	84,41%	78,91%
A11	Medição da densidade de defeito (defeitos / mil linhas de código).	61,65%	61,67%	61,66%
A12	Realização de treinamentos sobre testes.	76,84%	91,54%	85,69%
A13	Separação das atividades de teste do desenvolvimento.	84,71%	92,35%	89,31%
A14	Conservação dos dados dos testes para usos futuros.	74,09%	92,00%	84,87%
A15	Análise regular do padrão (tendência) de falhas e defeitos encontrados.	63,49%	82,20%	74,75%
A16	Existência de testadores em tempo integral para realização de testes.	74,07%	92,19%	84,98%
P1	Documentação do Plano de Teste descrevendo objetivos/abordagens.	89,65%	98,83%	95,17%
P2	Documentação de Procedimentos e Casos de Teste e seus Resultados.	82,71%	94,28%	89,67%
P3	Registro de falhas e defeitos detectados durante os testes.	86,33%	98,83%	93,85%
P4	Análise e medição da cobertura dos testes.	81,32%	86,38%	84,36%
P5	Utilização de metodologia ou processo para sistematizar e organizar testes de software.	87,45%	92,28%	90,36%
P6	Análise de defeitos identificados durante os testes para descobrir suas causas e como foram introduzidos (rastreamento).	74,67%	88,38%	82,92%
P7	Identificação e utilização de riscos para projetar, organizar e executar testes de software.	82,22%	74,60%	77,63%
P8	Desenvolvimento de testes de software antes da codificação.	77,29%	72,63%	74,48%
P9	Monitoração da aderência ao processo de teste de software.	72,12%	84,48%	79,56%
P10	Re-execução dos testes quando o software é modificado.	90,56%	95,11%	93,30%
P11	Avaliação dos documentos de teste (testware) quanto à sua qualidade.	80,84%	86,82%	84,44%
F1	Disponibilidade de um banco de dados de teste (pacote de testes que simulam o ambiente de teste) para reutilização.	80,53%	84,13%	82,69%
F2	Utilização de ferramentas geradoras de procedimentos/casos de testes.	83,43%	69,33%	74,95%
F3	Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste.	80,37%	87,12%	84,43%
F4	Utilização de ferramentas para estimar esforço de teste ou cronograma.	81,63%	78,38%	79,67%
F5	Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.	76,45%	82,43%	80,05%
F6	Utilização de ferramentas de medida de cobertura nas fases de teste (unidade, integração, sistema).	78,74%	78,16%	78,39%
MÉDIA		80,58%	86,97%	84,42%
DESVIO PADRÃO		80,30%	8,90%	7,68%
MEDIANA		79,98%	88,38%	84,43%

3.4 – Avaliação da Aplicabilidade x Grau de Importância

A partir da avaliação individual de cada objetivo do estudo, aplicabilidade e grau de importância, foi realizada uma análise combinada entre eles com o intuito de obter indicações sobre o estado da prática na área de teste de software, ou seja, o que é feito atualmente nas organizações e vem apresentando resultados positivos, as principais necessidades das organizações e atividades ou procedimentos que não são relevantes para essas organizações.

Com isso, os itens do questionário foram divididos em quatro grupos para análise quantitativa e qualitativa dos resultados do estudo: (1) *aplicados e importantes*; (2) *não-aplicados e importantes*; (3) *aplicados e não-importantes*; e (4) *não-aplicados e não-importantes*. Para isso, utilizou-se o seguinte critério:

Definição de Aplicabilidade:

Considera-se um item *aplicado* aquele que a soma da quantidade de organizações que obtiveram as escalas 3 (Uso Habitual) ou 4 (Uso Padrão) for maior que a soma das organizações que obtiveram as demais escalas de aplicabilidade. Caso contrário ela será considerado não-aplicada.

Definição de Importância:

Para definir se um item é considerado *importante*, obtiveram-se os quartis associados aos resultados do grau de importância (Tabela 3.10), conforme expresso na Tabela 3.11. A partir disso, somente os itens cujo grau de importância está acima do segundo quartil, ou seja, está acima da mediana, são considerados *importantes* neste estudo. Caso contrário, o item é considerado *não-importante*. A utilização da mediana garante que a metade das práticas será considerada importante.

Tabela 3.11. Divisão dos resultados obtidos em quartis.

Quartis da distribuição	Contexto Geral	Micro e Pequenas	Grandes	Classificação
Quartil 0 (valor mínimo da amostra)	até 61,66%	até 61,65%	até 61,67%	Não Importante
1º Quartil	até 79,56%	até 74,67%	até 82,43%	
2º. Quartil (mediana)	até 84,43%	até 81,32%	até 88,38%	
3º. Quartil	até 90,94%	até 87,01%	até 92,35%	Importante
4º. Quartil (valor máximo da amostra)	até 95,98%	até 92,34%	até 100,00%	

Essa forma de classificar as práticas de teste é descrita e utilizada em dois trabalhos científicos: VAN OSSEL (1998) e XEXÉO (2001). Neste segundo, utilizou-se classificação similar para entender a relação entre o nível de importância e de percepção de fatores críticos de qualidade em software médico.

A divisão desses grupos permitirá uma maior interpretação dos resultados obtidos e das reais necessidades e carências de organizações de software em relação às atividades de teste de software, seguindo o raciocínio abaixo:

- **APLICADO E IMPORTANTE:** isso indica que esse item avaliado vem apresentando resultados positivos quando aplicado nas organizações, de forma que os profissionais o julguem importante e deverão continuar a utilizá-lo em seus projetos. As pesquisas na área de teste de software devem prover meios para que ele continue a ser aplicado nas organizações.
- **APLICADO E NÃO-IMPORTANTE:** isso indica que esse item não vem apresentando resultados positivos quando aplicado nas organizações, de forma que as pessoas não o consideram importante em um projeto, apesar de usá-lo. A tendência é que esses itens sejam abandonados com o tempo, por não apresentarem resultados para a organização. As pesquisas devem buscar entendimento sobre a real necessidade destas práticas para as atividades de teste.
- **NÃO-APLICADO E NÃO-IMPORTANTE:** isso indica que esse item não faz parte dos planos das organizações para as atividades de teste, por não serem considerados importante. Os motivos relacionados a isso podem ser vários, como por exemplo, falta de conhecimento sobre o item avaliado ou por julgar que ele não faz parte do escopo dos projetos das organizações. As pesquisas na área de teste de software devem buscar mecanismos para aplicação destes itens em ambientes reais de desenvolvimento, para avaliar a sua importância a partir dos resultados de sua implantação nesses ambientes.
- **NÃO-APLICADO E IMPORTANTE:** isso indica que esse item, apesar de ser considerado importante, não está sendo aplicado nas organizações. Os motivos podem ser diversos, carência de recursos humanos, falta de apoio da gerência, falta de apoio ferramental, dentre outros. Esses itens indicam a necessidade imediata das organizações em relação às atividades de teste de software. As pesquisas devem buscar mecanismos para apoiar a sua aplicação em ambientes reais de desenvolvimento.

3.4.1 – Aplicabilidade x Grau de Importância no Contexto Geral

A partir da definição do critério para agrupamento dos itens, inicialmente avaliaram-se todos os itens em um contexto geral, ou seja, sem a divisão da amostra em grupos de acordo com o tamanho das organizações. Com isso, obteve-se os resultados apresentados na Tabela 3.12 e visualizados graficamente na Figura 3.7.

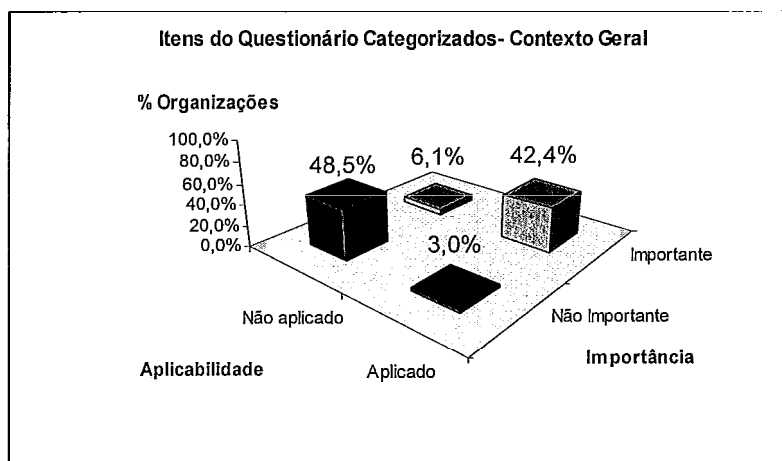


Figura 3.7. Avaliação de Aplicabilidade x Grau de Importância no contexto geral

Nesta tabela, as siglas (G),(O),(P),(C),(M) e (F) antes do título de cada prática de teste avaliada representam as suas categorias (Genérico, Organização, Planejamento, Controle, Medição e Análise e Ferramentas, respectivamente).

Tabela 3.12 – Divisão dos itens do questionário: Aplicabilidade x Importância.

	Não- Aplicado	Aplicado
Não-Importante	<ul style="list-style-type: none"> • (G) Realização de testes de regressão • (P) Identificação dos riscos de teste • (P) Desenvolvimento de teste antes da codificação • (C) Registro de tempo dos testes • (C) Monitoração da aderência ao processo de testes • (M) Todos os 5 itens de medição e análise (ver Tabela 3.6); • (F) Todos os 6 itens de ferramenta (ver Tabela 3.6); <p>TOTAL: 16 itens</p>	<ul style="list-style-type: none"> • (C) Conservação dos registros (log) dos testes executados; <p>TOTAL: 1 item</p>
Importante	<ul style="list-style-type: none"> • (O) Realização de treinamento sobre testes; • (O) Existência de testadores em tempo integral para realização de testes; <p>TOTAL: 2 itens</p>	<ul style="list-style-type: none"> • (G) Realização de testes de unidade, integração, sistema e aceitação • (O) Existência de um responsável ou equipe alocada para os testes • (O) Separação entre testes e desenvolvimento • (O) Utilização de metodologia para sistematizar os testes • (O) Re-execução dos testes quando o software é modificado • (P) Documentação do Plano de Testes • (P) Documentação de Procedimentos/Casos de Teste e seus resultados • (C) Registro das falhas e defeitos detectados • (C) Conservação dos dados dos testes para uso futuro • (C) Avaliação dos documentos de teste <p>TOTAL: 13 itens</p>

Avaliando-se os resultados obtidos, percebe-se que a maioria dos itens é considerada *não-aplicado e não-importante* (48,5%) ou *aplicado e importante* (42,4%). Para uma melhor interpretação dos resultados, os itens serão avaliados de acordo com a sua categoria.

- **Itens Genéricos:**

Entre esses itens, quatro são considerados *aplicados e importantes* pelas organizações participantes do estudo (*Realização de Teste de Unidade, Integração, Sistema e Aceitação*), porém eles não representam nenhum procedimento específico de apoio às atividades de teste de software. Para comprovação de que esses itens são aplicados nas organizações, é necessário avaliar os níveis abaixo da hierarquia dos itens descrita na seção 3.2.4. O outro item desse grupo (*Realização de Teste de Regressão*) é considerado *não-importante e não-aplicado* pela maioria das organizações participantes do estudo.

- **Itens de Organização:**

Entre esses itens, todos foram considerados *importantes* pelos participantes do estudo. Sendo que quatro dos seis itens desse grupo são aplicados nas organizações que participaram do estudo: *Existência de um responsável ou uma equipe alocada para as atividades de teste, Existência de testadores em tempo integral para realização de testes, Utilização de metodologia ou processo para sistematizar e organizar testes de software, Re-execução dos testes quando o software é modificado*.

Os demais itens (*Realização de treinamentos sobre testes e Separação das atividades de teste do desenvolvimento*) não são *aplicados* nas organizações, representando carência reconhecidas pelas organizações nesses aspectos. Não há uma política definida para treinamento dos profissionais em técnicas ou abordagens de teste. Além disso, não há uma separação entre os testes e o desenvolvimento, pois normalmente são alocadas pessoas da equipe do projeto para realizar os testes.

- **Itens de Planejamento:**

Entre esses itens, há uma divisão entre as suas categorizações. Dois entre os quatro itens são considerados *importantes e aplicados* pelas organizações (*Documentação do Plano de Teste descrevendo objetivos/abordagens e Documentação de Procedimentos e Casos de Teste e seus Resultados*). No entanto, os outros dois itens (*Identificação e utilização de riscos para projetar, organizar e executar testes de software e Desenvolvimento de testes de software antes da codificação*) são considerados *não-importantes e não-aplicados* nessas organizações. Isso indica que as organizações

realizam um planejamento dos testes, porém de forma limitada. Os documentos referentes ao planejamento dos testes são desenvolvidos, porém o momento em que os documentos são desenvolvidos e o conteúdo desses documentos pode não estar de acordo com as práticas de planejamento dos testes sugerida pela literatura técnica dos testes, como por exemplo, o IEEE *Standard 829* (1998).

- **Itens de Controle:**

Existe uma grande divergência entre esses itens, indicando uma limitação ou carência das atividades de controle dos testes. Existem itens em três dos quatro grupos de avaliação de aplicabilidade x grau de importância.

Entre esses itens, três são considerados *importantes* e *aplicados* pelas organizações: *Registro de falhas e defeitos detectados durante os testes*, *Conservação dos dados dos testes para usos futuros* e *Avaliação dos documentos de teste (testware) quanto à sua qualidade*. Dois itens foram considerados *não-importantes* e *não-aplicados* nas organizações: *Registro do tempo gasto em testes* e *Monitoração da aderência ao processo de teste de software*. Isso deve ocorrer porque muitas organizações não possuem um processo de teste formal a ser monitorado, e com isso não julgam tal prática importante.

A curiosidade é que um item foi considerado *não-importante* e *aplicado*: **Conservação de registro (log) dos testes executados**. Isso indica que os participantes do estudo não julgam este item como essencial para os testes realizados, apesar de aplicarem em seus projetos. Este resultado demonstra que as organizações não sabem o que fazer com os dados coletados nos testes. Isso ocorre provavelmente porque a conservação dos dados dos testes só tem sentido caso eles sejam utilizados na prevenção de futuros defeitos em novos projetos, o que ocorre a partir da adoção de um mecanismo eficaz de medição e análise.

- **Itens de Medição e Análise:**

Todos os itens dessa categoria foram considerados *não-aplicados* e *não-importante* pelas organizações que participaram do estudo. Isso indica uma carência das organizações nas atividades de medição e análise, pois nenhuma prática avaliada é utilizada pelas organizações, de forma que elas sentem dificuldade em julgar a sua importância para as atividades de teste. Isso indica a dificuldade das organizações em avaliar os resultados das suas atividades de teste, e com isso prover melhorias constantes ao longo do tempo.

- **Itens de Ferramentas:**

Todos os itens dessa categoria foram considerados *não-aplicados* e *não-importante* pelas organizações que participaram do estudo. Isso indica uma carência das organizações na implantação de ferramentas que apóiam a realização dos testes, possivelmente por não possuírem uma estrutura definida e organizada para as atividades de teste de software.

Ferramentas computacionais só permitem o apoio à determinada tarefa caso esta tarefa já esteja definida, seja de conhecimento de todos e exista um procedimento pré-estabelecido para sua utilização. Caso contrário, a organização estará tentando automatizar uma coisa ainda desconhecida por ela.

Após a avaliação global dos resultados, foi realizada uma análise onde se dividiu a amostra de acordo com o tamanho das organizações que participaram do estudo (micro, pequena ou grande) para avaliação dos seus dados. Com isso, a amostra foi dividida em dois grupos: (1) Micros e Pequenas Empresas e (2) Grandes Empresas, conforme já realizado anteriormente. Os resultados obtidos em cada amostra estão descritos nas seções seguintes.

3.4.2 – Aplicabilidade x Grau de Importância em Micros e Pequenas Empresas

De forma semelhante ao que foi descrito na seção 3.4.1, foi avaliado o relacionamento dos resultados de aplicabilidade e de grau de importância somente com os dados das 7 micros e pequenas empresas que participaram do estudo. A divisão dos itens está apresentada na Tabela 3.13 e representada graficamente na Figura 3.8.

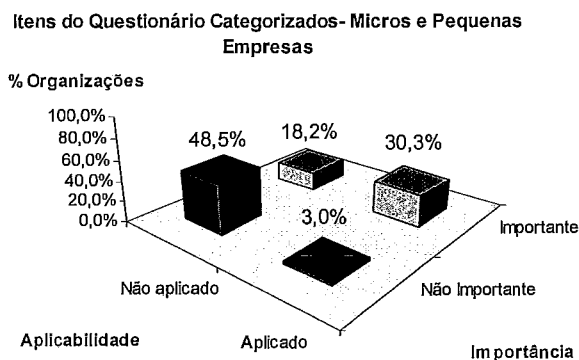


Figura 3.8. Avaliação de Aplicabilidade x Importância (Micros e Pequenas Empresas)

Estes resultados apresentam um cenário similar ao apresentado no contexto geral (seção 3.4.1). Foi obtido o mesmo número de itens considerados *não-importantes* e *não-aplicados* (16 itens), e são basicamente os mesmos itens, com pequenas alterações. De forma semelhante, todos os itens da categoria *Medição e Análise* são considerados *não-importantes* e *não-aplicados*. Além disso, o mesmo item (**Conservação de registro (log) dos testes executados**) foi isoladamente considerado *não-importante* e *aplicado* nas organizações que formam esta amostra. Porém, algumas variações puderam ser observadas:

Tabela 3.13 - Avaliação de Aplicabilidade x Importância (Micros e Pequenas Empresas)

	Não- Aplicado	Aplicado
Não-Importante	<ul style="list-style-type: none"> • (O) Realização de treinamento sobre testes • (O) Existência de testadores em tempo integral para realização de testes • (P) Desenvolvimento de testes antes da codificação • (C) Registro do tempo gasto em testes • (C) Conservação dos dados dos testes para uso futuro • (C) Monitoração da aderência ao processo de testes • (C) Avaliação dos documentos de teste quanto à sua qualidade • (M) Todos os 5 itens de medição e análise (ver Tabela 3.7) • (F) Disponibilidade de um banco de dados de teste (pacote de testes que simulam o ambiente de teste) para reutilização • (F) Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução dos testes • (F) Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos • (F) Utilização de ferramentas de medida de cobertura nas fases de teste <p>TOTAL: 16 itens</p>	<ul style="list-style-type: none"> • (C) Conservação de registro (<i>log</i>) dos testes executados <p>TOTAL: 1 item</p>
Importante	<ul style="list-style-type: none"> • (G) Realização de testes de regressão • (O) Separação dos testes do desenvolvimento • (O) Re-execução dos testes quando o software é modificado • (P) Identificação e utilização de riscos para projetar, organizar e executar testes de software • (F) Utilização de ferramentas geradoras de procedimentos/casos de teste • (F) Utilização de ferramentas para estimar esforço de teste ou cronograma <p>TOTAL: 6 itens</p>	<ul style="list-style-type: none"> • (G) Realização de testes de unidade, integração, sistema e aceitação • (O) Existência de um responsável ou uma equipe alocada para as atividades de teste • (O) Utilização de metodologia ou processo para sistematizar e organizar testes de software • (P) Documentação do Plano de Testes • (P) Documentação de Procedimentos e Casos de Teste e seus resultados • (C) Registro das falhas e defeitos detectados <p>TOTAL: 9 itens</p>

A quantidade de itens *aplicados* e julgados como *importante* reduziu bastante em relação à quantidade obtida no contexto geral (de 42,4% para 30,3%). Os itens que deixaram esse grupo são referentes às categorias *Organização* (são eles: separação entre os testes e o desenvolvimento e re-execução dos testes após modificações no produto) e *Controle* (são eles: conservação dos dados dos testes e avaliação da qualidade dos documentos de teste), revelando a limitação de micros e pequenas

empresas em relação à organização e controle dos testes, ocasionado possivelmente devido à carência de recursos humanos nessas organizações, o que impossibilita a definição de uma equipe específica responsável pelas atividades de testes (inclusive o seu controle).

Aliado a isso, a quantidade de itens considerados *importantes e não-aplicados* aumentou em relação ao contexto geral (de 6,1% para 18,2%). Os itens que ingressaram neste grupo são referentes ao planejamento dos testes e ferramentas de apoio à automação de tarefas de teste (são eles: geração de casos e procedimentos de teste e geração de estimativas de esforço e cronograma). Isso demonstra o interesse destas organizações em relação ao planejamento dos testes, que é pouco aplicado nos seus projetos, e na adoção de ferramentas de automação, como mecanismo para suprir a carência de recursos humanos responsáveis pelo planejamento dos testes (que é a fase onde é realizada a geração dos casos e procedimentos de teste e a estimativa de esforço e cronograma).

3.4.3 – Aplicabilidade x Grau de Importância em Grandes Empresas

A avaliação do relacionamento dos resultados de aplicabilidade e de grau de importância foi realizada também para os dados das 6 grandes empresas que participaram do estudo. A divisão dos itens está descrita na Tabela 3.14 e representada graficamente na Figura 3.9. Estes resultados apresentam um cenário diferenciado em relação ao contexto geral e em relação aos resultados de micros e pequenas empresas.

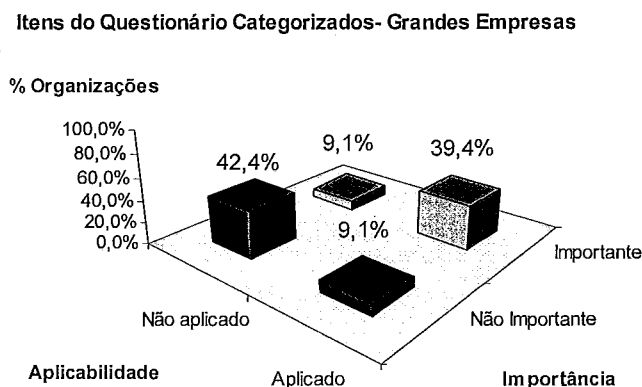


Figura 3.9. Avaliação de Aplicabilidade x Importância (Grandes Empresas)

Tabela 3.14 – Avaliação de Aplicabilidade x Importância (Grandes Empresas)

	Não- Aplicado	Aplicado
Não-Importante	<ul style="list-style-type: none"> • (G) Realização de testes de regressão • (P) Identificação dos riscos de teste • (P) Desenvolvimento de teste antes da codificação • (C) Monitoração da aderência ao processo de testes • (M) Medição dos testes e rastreamento dos custos com os testes • (M) Medição da densidade de defeitos • (M) Análise regular do padrão (tendência) de falhas e defeitos encontrados • (M) Análise de defeitos identificados durante os testes para descobrir suas causas e como foram introduzidos (rastreamento) • (F) Todos os 6 itens de ferramenta (ver Tabela 3.8) <p>TOTAL: 14 itens</p>	<ul style="list-style-type: none"> • (C) Conservação de registro (<i>log</i>) dos testes executados • (C) Avaliação dos documentos de teste quanto à sua qualidade • (M) Análise e medição da cobertura dos testes <p>TOTAL: 3 itens</p>
Importante	<ul style="list-style-type: none"> • (G) Realização de testes de unidade • (O) Realização de treinamento sobre testes • (C) Registro do tempo gasto em testes <p>TOTAL: 3 itens</p>	<ul style="list-style-type: none"> • (G) Realização de testes de integração, sistema e aceitação • (O) Existência de um responsável ou uma equipe alocada para as atividades de teste • (O) Separação das atividades de teste do desenvolvimento • (O) Existência de testadores em tempo integral para realização de testes • (O) Utilização de metodologia ou processo para sistematizar e organizar testes de software • (O) Re-execução dos testes quando o software é modificado • (P) Documentação do Plano de Testes • (P) Documentação de Procedimentos e Casos de Teste e seus resultados • (C) Registro das falhas e defeitos detectados • (C) Conservação dos dados dos testes para uso futuro <p>TOTAL: 12 itens</p>

Nesta amostra, a quantidade de itens *aplicados* é bem maior que em micros e pequenas empresas (48,5% contra 33,3%), o que comprova a hipótese de que grandes empresas aplicam mais práticas de teste de software em seus projetos, possivelmente por desenvolverem softwares mais complexos e terem mais recursos disponíveis.

Em relação aos itens *aplicados e importantes*, praticamente todos os itens da categoria *Organização* fazem parte deste grupo (5 itens entre os 6), o que indica que grandes empresas possuem uma maior preocupação em organizar formalmente os testes que micros e pequenas por trabalharem com equipes maiores, de forma que é necessária a divisão de responsabilidades para facilitar a comunicação entre as partes envolvidas.

Nesta amostra, 3 itens são considerados *não-importantes*, mas são *aplicados* nessas organizações: **Conservação de registro (log) dos testes executados, Avaliação dos documentos de teste quanto à sua qualidade e Análise e medição da cobertura dos testes**. Esses itens estão relacionados ao armazenamento, medição e análise dos dados dos testes, indicando uma deficiência das organizações na aplicação

dessas práticas, pois o que está sendo aplicado atualmente não está apresentando resultados para a organização.

Apesar disso, alguns resultados foram similares aos obtidos no contexto geral. Os itens considerados *não-aplicados e não-importantes* estão relacionados principalmente à medição e análise e utilização de ferramentas de apoio aos testes. Já os itens considerados *importantes e não-aplicados* estão relacionados à realização de testes de unidade, realização de treinamentos sobre testes e registro do tempo gasto em testes.

3.4.4 – Avaliação de Micros e Pequenas Empresas X Grandes Empresas

Após a análise de cada categoria de organização, podem ser observados pontos em comum, ou seja, coisas necessárias em ambos os contextos, e coisas específicas para uma categoria de organização. Uma sumarização dessas observações está descrita na Tabela 3.15.

Tabela 3.15 – Pontos comuns e específicos nas diferentes categorias de organização

Pontos em Comum	Pontos Específicos
<ul style="list-style-type: none"> • Não utilização de práticas de medição e análise nas suas atividades de teste; • Não utilização de ferramentas de apoio à realização de tarefas de teste de software; • Registro das falhas que ocorrem no produto durante os testes; • Documentação do plano, casos e procedimentos de teste, porém com o conteúdo limitado; • Realização de testes de sistema, integração e aceitação em seus projetos, porém, de forma limitada; • Conservação dos dados dos testes, porém sem utilizá-los posteriormente a fim de prover melhorias nos testes. Desta forma, não se obtém resultados com a conservação dos dados, e com isso a prática não é considerada importante para as organizações; • Limitação na aplicação de planejamento e controle dos testes. • Não aplicação de treinamentos constante com seus profissionais; 	<ul style="list-style-type: none"> • Grandes empresas julgam o treinamento dos profissionais em relação a práticas de teste como importante. Micros e pequenas empresas, não; • Grandes empresas possuem uma maior preocupação com a organização dos testes, criação de equipes específicas de teste, inclusive por possuírem mais recursos humanos disponíveis; • Grandes empresas não aplicam frequentemente testes de unidade em seus projetos, pois estão preocupadas com algo mais amplo; • Micros e pequenas empresas possuem o desejo de utilizarem ferramentas que apóiam o planejamento dos testes, apoiando às estimativas de teste e geração de casos e procedimentos de teste como forma de suprir a carência de recursos humanos; • Micros e pequenas empresas não julgam o registro do tempo gasto nos testes como importante para os testes;

3.5 – Conclusão

Neste capítulo foi apresentado o planejamento, a execução e os resultados obtidos a partir de uma pesquisa de opinião cujo objetivo era caracterizar práticas de teste de software em ambientes reais de desenvolvimento. Para isso, avaliaram-se dois aspectos: a *aplicabilidade* dessas práticas nas organizações de software, e o *grau de importância* das práticas na visão de pessoas que trabalham diariamente com o desenvolvimento de software.

Os resultados deste estudo não podem ser generalizados para outros cenários de desenvolvimento, mas permite observar que no cenário avaliado, as organizações possuem limitações quanto à aplicação de práticas de teste em seus projetos. Uma maior preocupação é dada à organização dos testes, enquanto que as atividades de planejamento, controle, medição e análise são pouco aplicadas na indústria de software. Observou-se também que ferramentas de apoio aos testes (relacionadas à gerência ou automação dos testes) não estão sendo aplicadas nessas organizações.

Avaliou-se também o cenário dos testes de software de acordo com o tamanho dessas organizações: micros, pequenas e grandes empresas. Com isso, observou-se que grandes empresas aplicam mais práticas de teste em seus projetos, principalmente relacionadas à organização dos testes. Por outro lado, micros e pequenas empresas possuem um maior interesse em implantar ferramentas que apóiam a automação de tarefas relacionadas ao planejamento dos testes (que é pouco aplicado nesse cenário), como forma de suprir a sua carência em recursos humanos.

Além disso, foi observada uma grande quantidade de práticas consideradas *não-importantes e não-aplicadas*, independentemente do tamanho das organizações. As razões para isso podem ser diversas, tais como:

- Falta de interesse da organização em relação a essas práticas de teste;
- Falta de conhecimento técnico por parte dessas pessoas em relação à atividade de teste de software;
- Falta de recursos humanos disponíveis para a realização de determinadas práticas de teste;
- Falta de apoio da alta gerência na implantação de melhorias nas atividades de teste de software;

- Carência em abordagens, técnicas de teste ou ferramentas que apoiem essas práticas e que atendam às reais necessidades das organizações.

Com isso, neste Capítulo foram apresentadas indícios relacionados ao estado da prática das atividades de teste de software em um cenário de desenvolvimento de software. Os resultados revelam as limitações e necessidades dessas organizações, e motivam pesquisas na área de teste de software, principalmente relacionadas à definição de estratégias de apoio ao planejamento, controle, medição e análise dos testes, além da construção de infra-estruturas computacionais que visam apoiar a estas atividades.

Dessa forma, este trabalho descreveu observações relacionadas ao estado da arte (Capítulo 2) e ao estado da prática (este capítulo) das atividades de planejamento e controle de teste de software, caracterizado em um cenário de desenvolvimento de software específico, observando as principais carências e limitações nessa área em ambos os mundos: o científico e o industrial.

O Capítulo seguinte, "*Proposta de Apoio ao Planejamento e Controle de Teste de Software*", descreve uma abordagem (e seus elementos) visando a apoiar o planejamento e controle de teste de software que será utilizada neste trabalho.

Capítulo 4

Proposta de Apoio ao Planejamento e Controle de Testes de Software

Neste capítulo é apresentada a proposta de uma abordagem que apóia o Planejamento e Controle de Testes de Software que será seguida ao longo deste trabalho. São identificados e especificados os elementos que compõem a abordagem e que são necessários para a realização do planejamento e controle dos testes de software.

4.1 – Introdução

Ao longo deste trabalho descreveu-se a importância das atividades de planejamento e controle durante o desenvolvimento de software, como pode ser observado inclusive nos modelos de referência de processo de software, como CMMI (2000) e mps.BR (2005), e a dificuldade em identificar estudos científicos que descrevam abordagens para apoiar exclusivamente o planejamento e controle de testes de software.

Aliado a isso, os resultados obtidos após a realização da pesquisa de opinião para caracterização inicial do estado da prática da área de teste de software em um cenário de desenvolvimento de software específico, descrita no Capítulo 3, possibilitou-se observar carências, limitações e o interesse de organizações de software localizadas em um pólo de desenvolvimento de software brasileiro na realização das atividades de planejamento e controle de testes.

Com isso, foi definida uma abordagem indicando os elementos que são relevantes para a realização do planejamento e controle de testes de software.

Neste Capítulo será descrita a abordagem proposta e os elementos que a compõem, visando ao apoio às atividades de planejamento e controle de testes de software. Sendo assim, na Seção 4.2 é apresentada uma descrição desta abordagem e seus elementos: **documentação** e **sistematização**, prevendo a construção de uma infraestrutura computacional para apoiar o planejamento e controle de testes de software.

Na Seção 4.3 é descrito o IEEE *Standard* 829 (1998), que consiste na estratégia adotada neste trabalho para **documentação** dos testes (que corresponde ao primeiro elemento que compõe a abordagem proposta). Na Seção 4.4 é apresentada a definição detalhada de um processo de testes de software a ser seguido neste trabalho, como estratégia adotada para **sistematização** dos testes (que corresponde ao segundo elemento que compõe a abordagem proposta). Por fim, na seção 4.5 são descritas as conclusões obtidas neste capítulo.

4.2 – Descrição da Abordagem e seus Elementos

A abordagem definida visa à descrição dos elementos necessários para a realização do planejamento e controle de atividades de teste de software, sendo composta por dois elementos básicos: **documentação** e **sistematização** (DIAS NETO e TRAVASSOS, 2005a), e foi baseada na metodologia descrita em (CRESPO *et al.*, 2004). A seguir, será detalhado cada elemento que compõe a abordagem, de acordo com a estrutura definida na Figura 4.1.

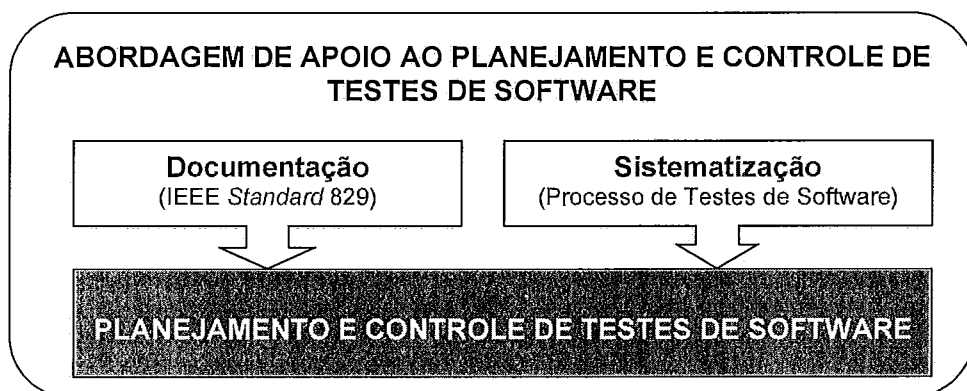


Figura 4.1. Composição da Abordagem de Apoio ao Planejamento e Controle de Testes de Software

4.2.1 – Documentação dos Testes

A documentação tem como objetivo principal fornecer evidências (registros) sobre a realização de determinada atividade e facilita a comunicação entre as diferentes partes envolvidas. Este elemento da abordagem proposta contribui para o planejamento dos testes por meio da especificação antecipada do que deve ser feito, quando deve ser feito e quem deve fazer cada tarefa, enquanto para o controle dos testes, a documentação

permite uma comparação posterior entre o que foi planejado previamente e o que ocorre efetivamente durante os testes por meio do registro das tarefas realizadas ao longo das atividades controladas e os resultados obtidos.

Como mecanismo adotado neste trabalho para documentação dos testes, será utilizado o IEEE *Standard* 829 (1998) – Padrão para Documentação dos Testes de Software, que consiste em um conjunto de documentos a serem produzidos ao longo das atividades de teste em um projeto de software. Esse padrão será descrito detalhadamente na seção 4.3 e foi escolhido por ser um padrão internacional, é gratuito e possui na sua estrutura documentos que possibilitam a documentação do planejamento, execução e análise dos resultados dos testes.

4.2.2 – Sistematização dos Testes

A sistematização tem como objetivo principal organizar o conjunto de atividades ou tarefas a serem realizadas durante os testes. A partir disso, diversos benefícios podem ser obtidos, tais como:

- Definição do conjunto de atividades a serem realizadas durante os testes, e a ordem em que devem ser executadas;
- Definição de papéis e responsabilidade para cada atividade a ser realizada;
- Definição de prazos e metas a serem atingidas a cada momento durante os testes;
- Definição dos artefatos a serem produzidos ao longo das atividades de teste, e o momento da sua produção;
- Melhor alocação dos recursos definidos para o projeto, e;
- Gerenciamento da equipe de teste para um determinado projeto.

Como mecanismo para sistematização dos testes adotado neste trabalho, foi definido um processo de testes de software que será responsável por sistematizar as atividades de teste em uma organização e direcionará a construção dos documentos descritos no IEEE *Std* 829 (1998), apoiando, assim, o planejamento e controle dos testes (CRESPO *et al.*, 2004).

O processo de testes de software foi definido seguindo uma abordagem de modelagem de processos organizacionais apresentada em (VILLELA, 2004), que permite a representação gráfica dos itens de um processo. Esta notação gráfica está descrita parcialmente no Anexo A deste trabalho. O processo de testes de software especificado será apresentado detalhadamente na seção 4.4.

4.3 – IEEE *Standard* 829: Padrão para Documentação dos Testes

O IEEE *Standard* 829-1998 (1998) descreve um conjunto de documentos para as atividades de preparação, execução e registro dos testes em um software. Os oito documentos definidos pelo padrão que cobrem as tarefas de teste são: **Plano de Teste**, **Especificação de Projeto de Teste**, **Especificação de Procedimento de Teste**, **Histórico dos Testes**, **Relatório de Incidente de Teste**, **Relatório de Resumo de Teste**, **Relatório de Encaminhamento de Item de Teste**. O roteiro de cada um dos documentos descritos no IEEE *Std* 829 utilizado neste trabalho está disponível no Apêndice C, e a Figura 4.2 apresenta a relação entre esses documentos, segundo o IEEE *Std* 829 (1998). Os documentos são descritos, a seguir, de acordo com (CRESPO *et al.*, 2004).

- **Plano de Teste:** Apresenta o planejamento para execução dos testes, incluindo o objetivo, escopo, abordagem de teste a ser seguida, recursos físicos e humanos e cronograma das atividades de teste. Identifica os itens e funcionalidades a serem testadas, as características dos itens que deverão ser testadas, tarefas a serem realizadas e os riscos associados às atividades de teste.
- **Especificação de Projeto de Teste:** Refina a abordagem de teste a ser seguida e que foi apresentada no Plano de Teste para avaliação de um (ou vários) item de teste em relação a uma característica (ou uma combinação de características) de teste que tenha sido especificada no Plano de Teste. Este documento também identifica os casos e os procedimentos de teste, se existirem, e apresenta os critérios para aprovação do(s) item (ou itens) de teste avaliado(s) neste projeto de teste específico. Cada característica (ou combinação de características) a ser avaliada durante os testes deve possuir um documento de *Especificação de Projeto de Teste* específico.
- **Especificação de Caso de Teste:** Define um caso de teste, incluindo dados de entrada, resultados esperados, ações e condições gerais para a sua execução. Cada caso de teste identificado nos documentos de *Especificação de Projeto de Teste* deve possuir um documento de *Especificação de Caso de Teste*.
- **Especificação de Procedimento de Teste:** Especifica os passos para executar um ou um conjunto de casos de teste. Cada procedimento de teste identificado nos documentos de *Especificação de Projeto de Teste* deve possuir um documento de *Especificação de Procedimento de Teste*.

- **Histórico dos Testes:** Apresenta registros cronológicos dos detalhes relevantes relacionados à execução dos testes.
- **Relatório de Incidente de Teste:** Documenta qualquer evento anormal, ou seja, que não era esperado pelo testador, que ocorra durante a execução dos casos e procedimentos de teste e que requeira análise posterior.
- **Relatório de Resumo de Teste:** Apresenta de forma resumida os resultados obtidos durante os testes e provê avaliações baseadas nesses resultados.
- **Relatório de Encaminhamento de Item de Teste:** Identifica os itens encaminhados para teste no caso de equipes distintas serem responsáveis pelas tarefas de desenvolvimento e de teste. Este documento não fará parte do escopo deste trabalho por ser fazer parte exclusivamente do processo de testes, mas sim de um processo em um nível mais amplo que envolve desenvolvimento e testes.

Mais do que apresentar um conjunto de documentos que deve ser utilizado ou adaptado para determinadas organizações ou projetos, este padrão apresenta um conjunto de informações necessárias para a realização de testes em software. Sua correta utilização auxiliará a gerência a se concentrar tanto com as fases de planejamento e projeto quanto com a fase de execução dos testes, evitando que o início das atividades de testes ocorra somente após a conclusão da fase de codificação.

No entanto, este documento não especifica os passos a serem seguidos para o seu preenchimento, ou seja, a diretriz/sistemática para realização dos testes, o que resulta na necessidade de um mecanismo de sistematização para direcionar o preenchimento desses documentos e realização das atividades de teste. A sistematização ocorrerá por meio da definição de um processo de testes de software.

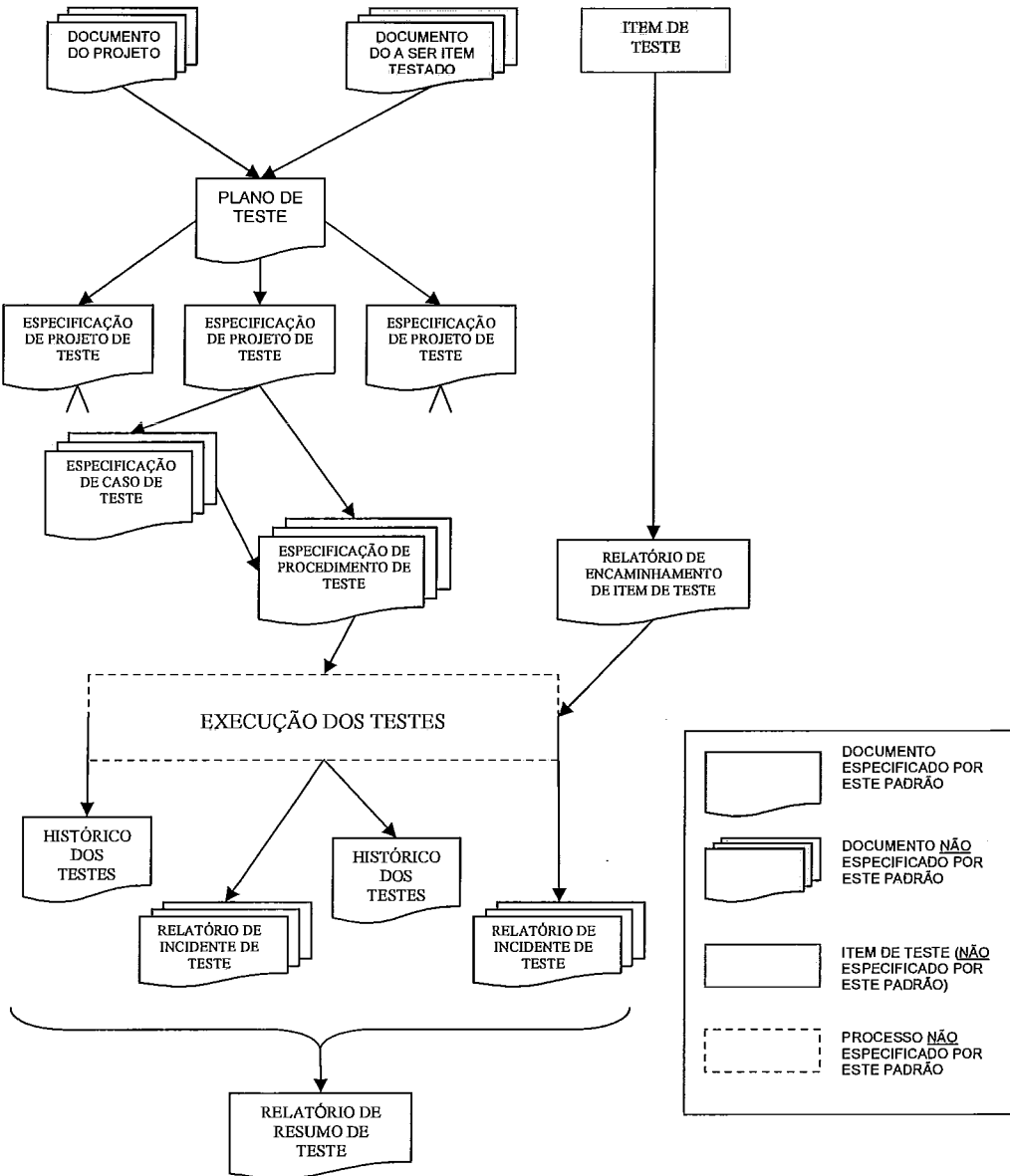


Figura 4.2. Relação entre os documentos descritos no IEEE Std 829 (1998)

4.4 – Definição do Processo de Testes de Software

O objetivo com a definição de um processo de testes é propor um mecanismo para a realização de forma sistemática das atividades de teste e que possa ser aplicado em

organizações de desenvolvimento de software ou adaptado de acordo com as necessidades e características dessas organizações.

Dessa forma, as atividades do processo de testes foram definidas baseando-se nas recomendações do IEEE *Std* 829 (1998), atividades descritas na literatura de teste software e de atividades de gerência de projetos de software, obtidas a partir de diversas fontes de conhecimento, como CMMI e PMBOK. Já os artefatos produzidos ao longo do processo de testes são os documentos especificados pelo IEEE *Std* 829, descritos na seção 4.3.

Além disso, o processo de testes de software deve ter papéis específicos em relação ao processo de desenvolvimento, de acordo com informações extraídas do SWEBOK – *Software Engineering Body of Knowledge* – (ABRAN *et al.*, 2001).

4.4.1 – Requisitos do Processo de Testes de Software

O processo de testes de software envolve todos os passos necessários ao planejamento, execução e controle das atividades de teste ao longo do processo de desenvolvimento de software. O objetivo a ser alcançado com a definição de um processo de testes de software é a especificação das tarefas, artefatos a serem produzidos e consumidos e os responsáveis por cada tarefa definida, possibilitando a sua implantação, por meio de uma adaptação inicial, em organizações de software.

Alguns requisitos fundamentais para o processo de testes a ser definido foram definidos a partir da caracterização das práticas de teste de software, obtida com o resultado da pesquisa de opinião descrita no Capítulo 3 deste trabalho, e algumas fontes de conhecimento da literatura de Engenharia de Software, e são os seguintes:

- **REQUISITO 1:** As macro-atividades do processo de testes a ser definido foram originadas do processo de testes apresentado por PFLEEGER (2004), e são: *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.*
- **REQUISITO 2:** De acordo com o SWEBOK (ABRAN *et al.*, 2001), a documentação dos produtos gerados ao longo de um processo é uma parte integral da sua formalização, pois indicam os artefatos que o compõem. O IEEE *Std* 829 define oito documentos para a documentação dos testes de software, que devem ser

utilizados como os artefatos produzidos ao longo do processo de testes a ser definido.

- **REQUISITO 3:** O processo deve atender às recomendações do padrão publicado pelo IEEE *Std 829* (1998) para produção dos seus artefatos no que diz respeito à relação entre os documentos de teste (ou seja, a ordem na qual eles são gerados) e seu conteúdo (ou seja, eles devem conter as informações descritas no IEEE *Std 829*);
- **REQUISITO 4:** Utilizar o conhecimento disponibilizado pelo CMMI (2000) no que diz respeito a questões relacionadas à gerência de um processo de software;
- **REQUISITO 5:** Utilizar práticas extraídas das literaturas técnicas da área de teste de software para a descrição de algumas atividades que formam as macro-atividades “Projetar Testes”, “Especificar Casos de Teste”, “Definir Procedimentos de Teste” e “Executar Testes”. As atividades são: *criação de casos e procedimentos de teste, priorização de casos e procedimentos de teste, manter rastreabilidade entre casos de teste e requisitos do software, executar casos e procedimentos de teste e registrar incidentes ocorridos durante os testes.*
- **REQUISITO 6:** Associar papéis e responsabilidades independentes para as atividades de teste, pois segundo o SWEBOK (ABRAN *et al.*, 2001), a definição de um processo de testes requer a especificação de uma equipe de teste na organização.

Os papéis associados ao processo de testes são:

- **Gerente de Teste.** Responsável pelo planejamento e controle dos testes.
- **Projetista de Teste.** Responsável pelo projeto dos testes, incluindo seleção de abordagens de teste e identificação e especificação dos casos e procedimentos de teste.
- **Testador.** Responsável pela execução dos procedimentos de teste especificados a fim de detectar falhas no software e registro dos incidentes ocorridos durante os testes.

A definição de um processo apresenta os critérios de entrada e saída que devem ser considerados, os produtos consumidos ou gerados e os responsáveis ou envolvidos em cada atividade. A notação a ser utilizada na definição do processo de testes de software foi descrita em (VILLELA, 2004). FALBO (1998) descreve um processo de software como sendo composto pelos seguintes elementos:

- **Sub-processos:** são processos que fazem parte de um processo mais amplo (FALBO, 1998)(Figura 4.3).

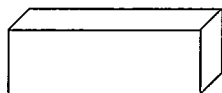


Figura 4.3. Representação gráfica de um processo ou sub-processo

- **Atividades:** são as tarefas ou trabalhos a serem realizados. Uma atividade requer recursos e pode consumir ou produzir artefatos. Uma atividade pode ser decomposta em outras atividades (atividade composta - Figura 4.4A) ou não (atividade atômica – Figura 4.4B). Além disso, atividades, em qualquer nível, podem depender da finalização de outras atividades, denominadas pré-atividades (FALBO, 1998). A dependência entre as atividades do processo são representadas por uma seta que une duas atividades (Figura 4.5).



Figura 4.4. Representação gráfica de uma (a) atividade composta e (b) atividade atômica

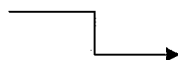


Figura 4.5. Representação gráfica da dependência entre atividade

- **Artefatos:** são produtos de software gerados ou consumidos por atividades do processo durante a sua realização (Figura 4.6). A seta tracejada indica o fluxo de entrada/saída, ou seja, uma ligação que estabelece um insumo ou um produto de uma atividade (FALBO, 1998).

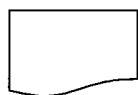


Figura 4.6. Representação gráfica de um artefato

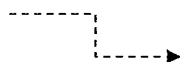


Figura 4.7. Representação gráfica do fluxo de entrada/saída

- **Recursos:** são as pessoas, as ferramentas de software, os equipamentos ou quaisquer outros recursos necessários à execução de uma atividade. Um

recurso humano, especificamente, desempenha um papel na execução das atividades do processo (FALBO, 1998) (Figura 4.8).



Figura 4.8. Representação gráfica de um papel associado ao processo

4.4.2 – Composição do Processo de Testes de Software

O processo de testes de software apóia a realização dos testes e fornece um guia para a equipe de teste, desde o seu planejamento até a avaliação dos seus resultados, de forma a garantir que os objetivos dos testes sejam atingidos (ABRAN *et al.*, 2001). Seu objetivo é minimizar os esforços e custos da realização dos testes por meio de um planejamento prévio e do seu acompanhamento intenso a fim de prover os melhores resultados nos testes realizados, e permitindo, ainda, reaproveitar as informações obtidas ao longo dos projetos anteriores em novos projetos.

A descrição detalhada do processo de testes de software definido, seus sub-processos e as macro-atividades e atividades que os compõem, com as pré-atividades e pós-atividades, critérios de entrada e saída, responsáveis, envolvidos, artefatos produzidos e consumidos, pode ser obtida em um Relatório Técnico do Programa de Engenharia de Sistemas e Computação (PESC) da COPPE/UFRJ. Esse Relatório Técnico pode ser obtido contatando os pesquisadores responsáveis por este trabalho.

A seguir são listadas as macro-atividades e atividades do processo de testes de software proposto, que é subdividido nos sub-processos de *Planejamento dos Testes* e *Execução dos Testes* (Figura 4.9):

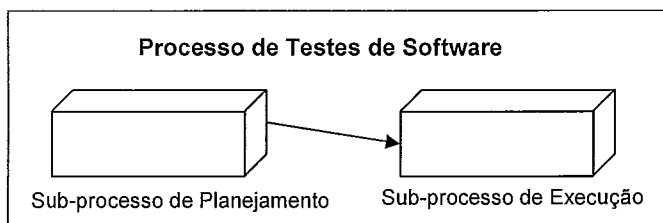


Figura 4.9. Sub-processos que compõem o processo de testes

4.4.2.1 – Sub-processo de Planejamento dos Testes

Descrição:

O objetivo deste sub-processo é planejar e definir o que será feito ao longo dos testes antes que comece a sua realização efetiva, ou seja, definir objetivos, cronograma, pessoas, riscos, casos e procedimentos de teste, dentre outros aspectos. As atividades desse processo produzem, ao final, os seguintes documentos: *Plano de Teste*, *Especificação de Projeto de Teste*, *Especificação de Caso de Teste* e *Especificação de Procedimento de Teste*, além dos casos de teste construídos, quando pertinente.

Macro-Atividades:

O sub-processo de Planejamento dos Testes é composto por 4 macro-atividades: *Planejar Teste*, *Projetar Testes*, *Especificar Casos de Teste* e *Definir Procedimentos de Teste* (Figura 4.10).

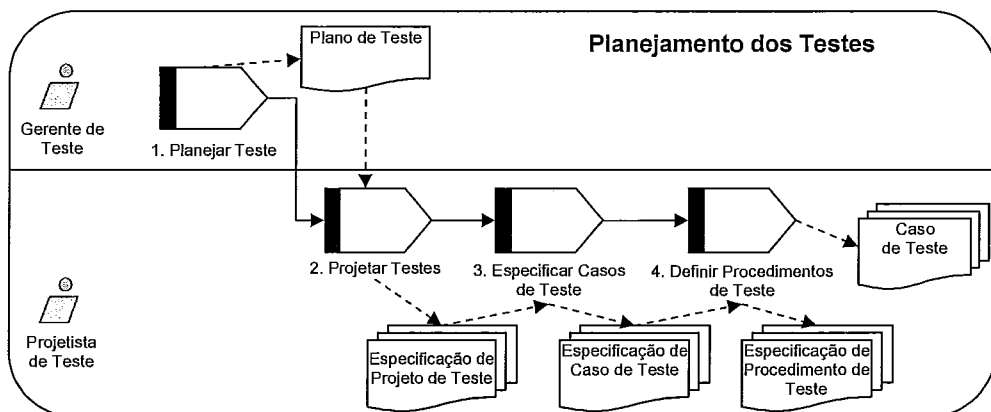


Figura 4.10. Sub-processo de *Planejamento de Testes*

Apesar da representação gráfica descrita na Figura 4.10 indicar a dependência entre as macro-atividades do sub-processo, há a possibilidade de paralelismo entre elas, respeitando as suas dependências, ou seja, a macro-atividade *Projetar Testes* só pode iniciar após o início da macro-atividade *Planejar Testes*, mas não há a necessidade que essa macro-atividade esteja concluída. O mesmo ocorre com as demais macro-atividades. A Figura 4.11 expressa um exemplo da temporização entre as macro-atividades do sub-processo de planejamento de testes, indicando como elas podem ser realizadas simultaneamente, respeitando suas dependências. Ela indica que o início (ou fim) de cada macro-atividade só pode ocorrer caso a macro-atividade anterior já esteja iniciada (ou concluída).

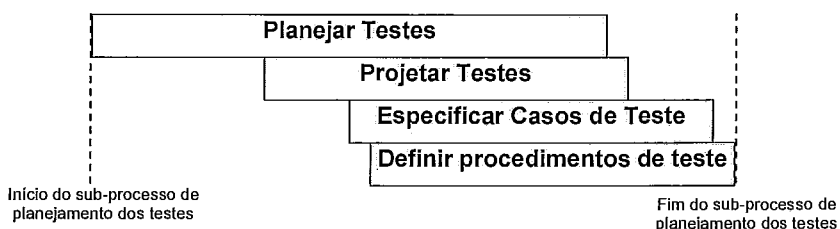


Figura 4.11. Temporização entre as macro-atividades do sub-processo *Planejamento de Testes*

Macro-Atividade 1. Planejar Teste

Planejamento é o ponto chave para o sucesso dos testes em um produto. O planejamento dos testes deve fazer parte do planejamento global do software, culminando em um Plano de Teste que constitui um dos documentos cruciais no ciclo de vida de desenvolvimento de software (GELPERIN e HETZEL, 1988). Ao longo dessa atividade é feito o planejamento do processo de teste a ser seguido para um projeto específico, onde são estimados custos, cronograma e recursos, e são definidos os itens a serem testados, as estratégias, métodos e técnicas de teste a serem adotadas, caracterizando-se um critério para aceitação do software testado.

Esta macro-atividade deve ser realizada pelo Gerente de Testes e envolve muitas tomadas de decisões que devem ser analisadas buscando-se encontrar as melhores opções, dado o contexto de um projeto. Pontos chaves do planejamento dos testes estão relacionados à forma como os testes serão realizados, qual a equipe de teste, os recursos disponíveis, restrições de tempo e os riscos associados (ABRAN *et al.*, 2001).

Atividades:

Atividade: *Caracterizar Testes.*

Durante esta atividade, os testes a serem realizados devem ser caracterizados. Neste momento serão especificadas as seguintes informações: Introdução, Itens de teste, Características do produto a serem testadas (como interface, performance, segurança ou funcionalidade) e Características que não serão testadas.

Atividade: *Planejar Estratégia para Testar o Produto.*

Durante esta atividade, serão especificadas as tomadas de decisões realizadas pelo gerente de teste em relação à escolha de técnicas, ferramentas e critério para aceitação 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, Critério de aprovação/rejeição de um item de teste.

Atividade: *Definir Atividades de Teste.*

Identificar o conjunto de atividades, suas relações e artefatos necessários para preparar (planejar) e executar os testes.

Atividade: *Planejar Recursos Humanos.*

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.

Atividade: *Planejar Recursos Físicos.*

O gerente de teste, baseando-se nas necessidades para a criação de um ambiente para projeto e execução dos testes para o projeto, deve realizar o planejamento 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.

Atividade: *Planejar Tempo para Teste.*

O gerente de teste, baseando-se planejamento realizado para o projeto e nas atividades definidas anteriormente, deve realizar o planejamento de tempo para os testes, indicando as data de início e conclusão para cada atividade do processo. Ao final, será obtido um cronograma dos testes no projeto corrente.

Atividade: *Planejar Custos.*

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.

Atividade: *Especificar Necessidade de Treinamento (opcional).*

Durante esta atividade, devem 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.

Atividade: *Identificar Riscos.*

Riscos são fatores fundamentais na realização de testes, pois indicam situações que podem ocorrer ao longo dos testes e afetar a qualidade de seus resultados. Durante esta atividade,

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.

Macro-Atividade 2. Projetar Testes

De acordo com o IEEE *Std* 829 (1998), o projeto de teste de software visa à especificação mais detalhada das abordagens a serem seguidas durante a realização dos testes e que foram identificadas durante a macro-atividade "Planejar Teste" para avaliação dos itens de teste identificados na macro-atividade anterior. Além disso, essa macro-atividade é responsável pela identificação do conjunto de casos e procedimentos de teste a serem executados para avaliação do software.

As atividades que devem ser realizadas durante o projeto de teste de software devem ser conduzidas pelo Projetista de Testes e são baseadas nas informações que compõem o documento *Especificação de Projeto de Teste*, descrito no IEEE *Std* 829 (1998).

Atividades:

Atividade: *Selecionar Item de Teste e Características.*

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.

Atividade: *Descrever Estratégia para o Projeto de Teste.*

Durante esta atividade, devem ser especificadas com detalhes a estratégia descrita no plano de teste para avaliar o item de teste e a característica selecionados 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. Além disso, deverá ser identificada a forma como os resultados dos testes serão analisados.

Atividade: *Identificar Casos e Procedimento de Teste.*

Casos de Teste são pontos essenciais na execução dos testes em software. Testar exaustivamente ou testar todas as combinações de entradas possíveis é inviável (MYERS, 1979). Com isso, a seleção de casos de teste corretos é um fator essencial para o sucesso da atividade de teste. 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.

Atividade: *Priorizar Casos e Procedimentos de Teste.*

Segundo DUSTIN (2002), o planejamento de casos e procedimentos de teste deve ser baseado na ordem de prioridade. 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 (DUSTIN, 2002). Durante esta atividade deve ser especificada a ordem de prioridade dos casos e procedimentos de teste levando-se em consideração esses ou outros aspectos definidos.

Atividade: *Especificar Critério de Aprovação/Rejeição da Característica:*

Durante esta atividade, devem ser especificados os critérios a ser utilizado para determinar, exclusivamente, quando a característica (ou a combinação de características) selecionada para o projeto de teste será aprovada ou rejeitada.

Macro-Atividade 3. *Especificar Casos de Teste*

Segundo CRAIG e JASKIEL (2002), um caso de teste descreve uma condição particular a ser testada e é definido por valores de entrada e um resultado esperado. Nessa macro-atividade, devem ser especificados todos os casos de teste identificados ao longo da macro-atividade anterior (Projetar Testes), isto é, devem ser descritos os seus valores de entrada, resultados esperados, recursos necessários para a sua execução, suas restrições e dependências com outros casos de teste.

As atividades que devem ser realizadas durante a especificação dos casos de teste devem ser conduzidas pelo Projetista de Teste e são baseadas nas informações que compõem o documento *Especificação de Caso de Teste*, descrito no IEEE Std 829 (1998).

Atividades:

Atividade: *Caracterizar Caso de Teste.*

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).

Atividade: *Especificar Entrada e Resultado Esperado.*

Casos de teste são compostos por valores de entrada e comportamentos e resultados esperados. Esta atividade define esses valores.

Atividade: *Definir Recursos para o Caso de Teste.*

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”.

Atividade: *Especificar Restrições de Uso.*

Durante esta atividade serão descritas possíveis restrições no caso de teste em questão. Essas restrições podem envolver configuração, procedimentos para obtenção dos resultados e conclusão do caso de teste.

Atividade: *Definir Dependências entre Casos de Teste.*

Identificar quais casos de teste devem ser executados antes do caso de teste em questão. Deve ser especificada a origem desta dependência.

Macro-Atividade 4. Definir Procedimento de Teste

Segundo CRAIG e JASKIEL (2002), um procedimento 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.

As atividades a serem realizadas durante a definição dos procedimentos de teste devem ser conduzidas pelo Projetista de Teste e são baseadas nas informações que compõem o documento *Especificação do Procedimento de Teste*, descrito no IEEE Std 829 (1998).

Atividades:

Atividade: *Definir Objetivos.*

Durante esta atividade, devem ser identificados os objetivos para um procedimento de teste, indicando quais itens de teste estão associados ao procedimento em questão.

Atividade: *Definir Requisitos para a Execução do Procedimento.*

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.

Atividade: *Descrever Passos.*

Durante esta atividade, devem ser descritos todos os passos a serem seguidos para a execução do procedimento de teste. Isto inclui diversas atividades, que são:

- 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 que ocorrem durante a execução dos testes.

Atividade: *Construir Procedimento de Teste (opcional).*

A partir de todas as informações definidas nas atividades anteriores, durante esta atividade o procedimento de teste deve ser construídos (por meio da criação de scripts), caso necessite, para serem utilizados durante o sub-processo de execução dos testes.

4.4.2.2 – Sub-processo de Execução dos Testes

Descrição:

O objetivo desse sub-processo é a execução dos testes a partir do que foi estabelecido durante o planejamento, monitorando as atividades realizadas, registrando os incidentes detectados e apresentando ao final os resultados dos testes.

Ao final deste sub-processo serão estabelecidos os seguintes documentos: *Histórico dos Testes*, *Relatório(s) de Incidente de Teste* e *Relatório de Resumo de Teste*, além do pacote com os dados dos testes realizados.

Macro-Atividades:

O sub-processo de Execução dos Testes é composto por 2 macro-atividades: *Executar Testes* e *Analisar Resultados* (Figura 4.12).

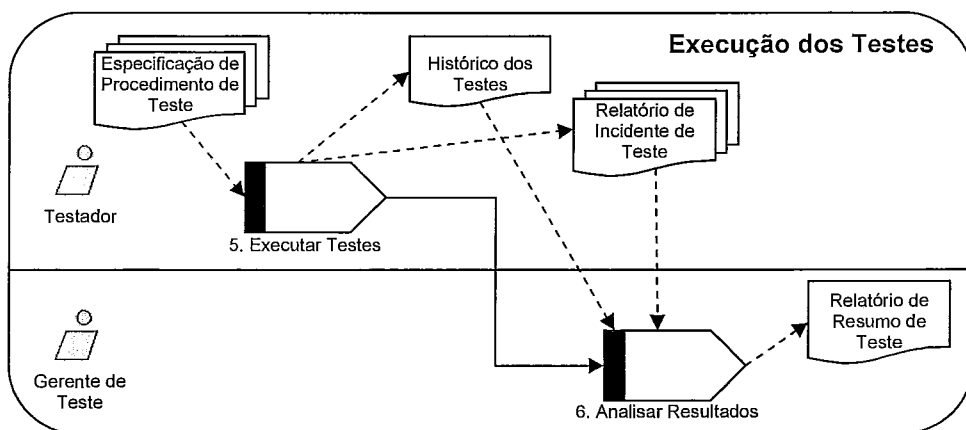


Figura 4.12. Sub-processo para *Executar Testes*

Macro-Atividade 5. Executar Testes

Segundo o SWEBOK (ABRAN *et al.*, 2001), a execução dos testes deve incorporar os princípios básicos de experimentação científica: tudo feito durante os testes deve ser realizado e documentado claramente permitindo que outras pessoas possam repetir os mesmos resultados. Portanto, os testes devem ser realizados de acordo com os procedimentos documentados para um produto específico.

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).

Atividades:

Atividade: *Configurar Ambiente de Teste.*

O ambiente de teste deve ser compatível com ambiente de desenvolvimento 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.

Atividade: *Executar Procedimentos de Teste.*

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*.

Atividade: *Registrar Incidentes de Teste.*

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.

Macro-Atividade 6. Analisar Resultados dos testes

Os resultados dos testes devem ser avaliados para determinar se os testes obtiveram sucesso, além de obter medidas de teste específicas (ABRAN *et al.*, 2001). Na maioria dos casos, “sucesso” significa que o sistema funcionou conforme o esperado, e não apresentou resultados inesperados. 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).

Atividades:

Atividade: *Resumir Testes.*

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.

Atividade: *Registrar Dados dos Testes.*

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.

4.5 – Conclusão

Neste Capítulo foi descrita uma abordagem que apóia o planejamento e controle de testes de software e os elementos que a compõem: *documentação* e *sistematização*. Para cada elemento, foi adotada uma estratégia a ser seguida ao longo deste trabalho. O apoio ao planejamento deve ocorrer através da execução das atividades do sub-processo Planejamento dos Testes que compõe o processo de testes de software definido e a produção de seus artefatos. Já o controle dos testes deve ocorrer através do registro dos dados ao longo do processo de testes, possibilitando o acompanhamento das atividades de teste. Isso pode garantir que os testes planejados sejam efetivamente executados, que as estratégias definidas durante o planejamento sejam seguidas, podem ser observados atrasos nos testes, o momento de finalizar os testes a partir dos critérios de aprovação/rejeição estabelecidos, os itens de teste mais propensos a falha, dentre outras informações.

A partir da sua definição, houve a possibilidade de aplicar a abordagem descrita neste Capítulo em um projeto de software real em uma organização pública brasileira. No entanto, os dados obtidos com a aplicação da abordagem proposta na indústria são de propriedade exclusivas da organização, confidenciais e não podem ser relatados neste trabalho.

O projeto ocorreu no período de maio a novembro de 2005, e possibilitou a observação da abordagem e os elementos definidos neste trabalho sendo utilizados na indústria, o que auxiliou na sua continuidade. Percebeu-se principalmente que apesar da definição de uma abordagem com os elementos que apóiam o planejamento e controle dos testes (*documentação* e *sistematização*), a implantação e gerenciamento de um processo complexo, que envolve várias pessoas, atividades e tarefas repetitivas de forma manual, ou seja, sem um apoio automatizado ou semi-automatizado, torna-se uma tarefa muito custosa em um projeto de software.

Com o objetivo de minimizar o esforço na realização da *documentação* e *sistematização* dos testes de software em um projeto, este trabalho propôs a construção de uma infra-estrutura computadorizada¹² que apóie a aplicação da abordagem de apoio ao planejamento e controle dos testes de software descrita neste Capítulo e em (DIAS

¹² Infra-estrutura consiste em uma base material que serve como alicerce para uma organização ou sociedade. Uma infra-estrutura computadorizada consiste em um programa computacional que auxilie na automatização ou semi-automatização de uma determinada tarefa ou processo.

NETO e TRAVASSOS, 2005b). Dessa forma, foi criado um mecanismo para automatização e gerenciamento das atividades do processo de testes de software e da construção dos diversos artefatos a serem produzidos ao longo do processo.

No próximo Capítulo, "*Maraká: Infra-estrutura Computacional para Apoio ao Planejamento e Controle de Testes de Software*", será descrita a infra-estrutura computacional, denominada Maraká, que apoiará a execução do processo de testes de software definido neste trabalho, enfatizando as atividades de planejamento e controle, e construção dos artefatos a serem produzidos ao longo desse processo. Ou seja, para cada atividade do processo de testes de software descrito neste Capítulo, será especificada uma estratégia que apóie a sua realização, referindo-se ao planejamento e controle dos testes.

Capítulo 5

Maraká: Infra-estrutura Computacional para Apoio ao Planejamento e Controle de Testes de Software

Neste capítulo é apresentada a infra-estrutura computacional construída para apoiar a documentação e sistematização dos testes, seguindo a abordagem de apoio ao planejamento e controle de teste de software proposta no capítulo anterior, por meio da execução do processo e produção de relatórios e gráficos de acompanhamento. São descritos os seus requisitos, arquitetura, decisões de projeto e o funcionamento detalhado da infra-estrutura, denominada Maraká, fornecendo apoio ao planejamento e controle de testes de software.

5.1 – Introdução

A partir da descrição dos elementos que compõem a abordagem de apoio ao planejamento e controle de testes de software definida no Capítulo 4 deste trabalho, e devido à dificuldade na realização dessas tarefas de forma manual, o passo seguinte consiste na elaboração de uma infra-estrutura computacional que permita a documentação e sistematização das atividades de teste de software, enfatizando o apoio às atividades de planejamento e controle dos testes (DIAS NETO e TRAVASSOS, 2005b).

A infra-estrutura computacional proposta deve fornecer um apoio automatizado ou semi-automatizado para a realização das tarefas de sistematização e documentação dos testes de software em uma organização. Ela deve considerar os diferentes perfis associados às atividades de teste e os diversos artefatos produzidos ao longo dos testes. Deve ser ressaltado que o objetivo da infra-estrutura é apoiar o planejamento e controle dos testes, e não a sua execução.

Este capítulo descreve a solução computacional adotada como mecanismos de apoio ao planejamento e controle de teste de software. Sendo assim, inicialmente na Seção 5.2 são identificados os requisitos para a construção da infra-estrutura proposta

com o objetivo de fornecer apoio ao planejamento e controle dos testes de software. Em seguida, na Seção 5.3 são descritas as decisões de projeto tomadas para a construção da infra-estrutura, a sua solução arquitetural e seu projeto de alto nível.

Na Seção 5.4 é apresentada a infra-estrutura Maraká¹³ por meio de suas funcionalidades, provendo um apoio ao planejamento e controle dos testes de software. Por fim, na Seção 5.5 são discutidas as conclusões deste capítulo.

5.2 – Requisitos da Infra-estrutura Maraká

Os requisitos da infra-estrutura foram definidos visando à aplicação da abordagem de apoio ao planejamento e controle de testes de software definido no Capítulo 4 por meio da execução das atividades do processo de testes e da geração dos artefatos que compõem o processo. Além disso, foram definidos requisitos para apoiar o gerenciamento da equipe de teste de software e o armazenamento dos dados obtidos nos testes realizados.

Os requisitos para a infra-estrutura foram identificados a partir de duas fontes de conhecimento: (1) a literatura científica da área de teste de software, descrevendo uma visão teórica sobre o assunto, e (2) a partir dos resultados obtidos na pesquisa de opinião descrita no Capítulo 3, visando a apoiar algumas carências e necessidades relatadas pelas organizações de software que participaram do estudo em relação às atividades de teste de software, descrevendo uma visão prática das atividades de planejamento e controle de testes de software a partir da perspectiva de pessoas que atuam nessas organizações.

5.2.1 – Apoio à Documentação e Sistematização dos Testes de Software

O objetivo principal da infra-estrutura Maraká é fornecer apoio à documentação e sistematização dos testes de software, seguindo a abordagem apresentada no Capítulo 4. Isso representa um dos mecanismos para realização do planejamento e controle dos testes. Para isso, diversos requisitos foram identificados:

REQ 1: A infra-estrutura deve permitir o acesso aos dados obtidos durante a execução do processo de testes de software instanciado para um projeto de acordo com o papel (*Gerente de Teste, Projetista de Teste e Testador*) definido para o usuário no processo.

¹³ Maraká é um instrumento usado pelo Pajé em rituais indígenas para expulsar os males que assombram a tribo. No nosso contexto, os males seriam as falhas reveladas em um produto e que tentamos eliminá-las.

REQ 2: A infra-estrutura deve permitir a construção de todos os artefatos do processo de testes de software, de acordo com as suas atividades. Os artefatos foram descritos no Capítulo 4 e seus roteiros estão disponíveis no Apêndice C, e são: *Plano de Teste*, *Especificação de Projeto de Teste*, *Especificação de Casos de Teste*, *Especificação de Procedimento de Teste*, *Relatório de Incidentes de Teste*, *Histórico dos Testes* e *Relatório de Resumo dos Testes*.

REQ 3: A infra-estrutura deve permitir a execução das atividades do processo de testes de software se, e somente se, as atividades antecessoras a elas já estejam iniciadas.

REQ 4: A infra-estrutura deve permitir que o usuário (*Gerente de Teste*, *Projetista de Teste* ou *Testador*) preencha as informações de cada artefato do processo de testes de software somente por meio da execução das atividades desse processo, seguindo a descrição dos artefatos a serem produzidos em cada atividade que foi especificada no Capítulo 4. Dessa forma, em cada atividade do processo de testes devem estar indicadas explicitamente na infra-estrutura quais informações devem ser preenchidas.

Na seção 5.2.1.1 serão identificados os requisitos específicos para apoiar o sub-processo de planejamento dos testes.

5.2.1.1 – Requisitos Relacionados ao Planejamento dos Testes

Os requisitos relacionados ao planejamento dos testes indicam facilidades para produzir o plano de teste, casos e procedimentos de teste durante a execução de um processo de testes.

REQ 5: A infra-estrutura deve permitir que somente o *Gerente de Teste* execute a macro-atividade *Planejar Testes* (e suas atividades) e, dessa forma, produza o documento *Plano de Teste*.

REQ 6: A infra-estrutura deve permitir que o *Gerente de Teste* ou o *Projetista de Teste* execute a macro-atividade *Projetar Testes* (e suas atividades) e, dessa forma, produza o documento *Especificação de Projeto de Teste* para cada característica (ou combinação de características) do produto a ser avaliada durante os testes.

REQ 7: A infra-estrutura deve permitir que o *Gerente de Teste* ou o *Projetista de Teste* execute a macro-atividade *Especificar Casos de Teste* (e suas atividades) e, dessa forma, produza o documento *Especificação de Caso de Teste* para cada caso de teste identificado.

REQ 8: A infra-estrutura deve permitir que o *Gerente de Teste* ou o *Projetista de Teste* execute a macro-atividade *Definir Procedimentos de Teste* (e suas atividades) e, dessa forma, produza o documento *Especificação de Procedimento de Teste* para cada procedimento de teste identificado.

Na seção 5.2.1.2 serão identificados os requisitos específicos para apoiar o controle dos testes.

5.2.1.2 – Requisitos Relacionados ao Controle dos Testes

Os requisitos relacionados ao controle dos testes indicam facilidades para acompanhar os resultados obtidos durante a execução de um processo de testes, e podem ser relatórios ou gráficos de acompanhamento.

REQ 9: A infra-estrutura deve permitir que o *Gerente de Teste*, *Projetista de Teste* ou *Testador* registre os resultados da execução dos procedimentos de teste especificados, produzindo o documento *Histórico dos Testes*.

REQ 10: A infra-estrutura deve permitir que o *Gerente de Teste*, *Projetista de Teste* ou *Testador* registre os incidentes ocorridos ao longo da execução dos testes planejados, produzindo o documento *Relatório de Incidente de Teste*, para cada incidente ocorrido.

REQ 11: A infra-estrutura deve permitir que o *Gerente de Teste* registre a análise dos resultados dos testes, resumindo os resultados obtidos, descrevendo variações no planejamento e resultado das avaliações dos itens, produzindo o documento *Relatório de Resumo dos Testes*.

REQ 12: A infra-estrutura deve permitir que qualquer usuário visualize e imprima os dados dos artefatos produzidos em outras atividades de teste e que foram acompanhadas utilizando-se a infra-estrutura.

REQ 13: A infra-estrutura deve permitir que qualquer usuário visualize e imprima as informações do cronograma das atividades de teste nas quais está alocado.

REQ 14: A infra-estrutura deve permitir que qualquer usuário visualize e imprima as informações de qualquer artefato já produzido durante a execução do processo de testes nos quais está alocado.

REQ 15: A infra-estrutura deve permitir que o *Gerente de Teste* visualize os dados sobre as atividades de cada membro alocado para um processo de testes que esteja gerenciando.

REQ 16: A infra-estrutura deve prover a visualização dos dados obtidos na execução dos testes por meio de gráficos que indique o andamento e resultados dos testes. Os gráficos são:

- **Gráfico de Análise de Procedimentos de teste:** analisa os resultados dos procedimentos de teste definidos durante a execução de um processo de teste de software instanciado através de um gráfico de pizza agrupando de acordo com o seu resultado (aprovado, rejeitado ou não-executado).
- **Gráfico de Incidentes Detectados:** analisa, em um gráfico de linhas, o número de defeitos detectados durante os testes a cada dia, a fim de indicar um aumento ou diminuição do número de defeitos encontrados.
- **Gráfico de Gantt:** a infra-estrutura deve permitir que o usuário controle e acompanhe, por meio de um gráfico de gantt, a execução do processo de testes para cada atividade de teste que foi alocado. O controle da execução do processo é realizado por meio da visualização das datas de início e término (planejadas e reais) das macro-atividades e atividades que compõem o processo, além de suas pré-atividades.
- **Gráfico de Incidentes por Item de Teste:** analisa, em um gráfico de barras, o número de incidentes em cada item de teste analisado, a fim de indicar pontos com maior probabilidade de falhas no produto.

A seção 5.2.2 descreve outros requisitos identificados para a infra-estrutura e que não estão relacionados especificamente à documentação ou sistematização dos testes.

5.2.2 – Outros Requisitos da Infra-estrutura

REQ 17: A infra-estrutura deve permitir que o usuário com perfil de *Gerente de Teste* instancie um novo processo de testes de software a ser acompanhado através de Maraká.

REQ 18: A infra-estrutura deve permitir que o *Gerente de Teste* cadastre um novo ou edite os dados de um membro da equipe de teste utilizando a infra-estrutura.

REQ 19: A infra-estrutura deve permitir que um gerente de teste visualize e imprima os dados dos membros da equipe de teste de software cadastrados na infra-estrutura.

REQ 20: A infra-estrutura deve estar disponível por meio da Internet, facilitando o seu acesso e disponibilização, visto que várias pessoas poderão ter acesso a ela simultaneamente.

REQ 21: A infra-estrutura deve permitir que cada usuário altere os dados de sua conta de acesso à infra-estrutura (e-mail e senha de acesso).

REQ 22: A infra-estrutura deve estar disponível somente para os usuários que possuam contas cadastradas na infra-estrutura.

REQ 23: A infra-estrutura deve estar disponível nos idiomas português e inglês, mas deve possibilitar a adição de novos idiomas.

A seção seguinte (5.3) apresenta o projeto de alto nível da infra-estrutura e as decisões de projeto tomadas para o seu desenvolvimento.

5.3 – Arquitetura, Decisões de Projeto e Projeto de Alto Nível

Nesta seção são descritas a solução arquitetural proposta para atender aos requisitos identificados, as decisões de projeto tomadas para implementar a arquitetura proposta, e por fim, o projeto de alto nível da infra-estrutura.

5.3.1 – Solução Arquitetural

A partir dos requisitos da infra-estrutura descritos na seção 5.2, foi projetada uma solução arquitetural que atenda a esses requisitos. Com isso, definiu-se uma arquitetura representada por quatro componentes, conforme visualizado na Figura 5.1, que são:

- **Controle de Acesso:** Este componente é responsável por viabilizar o acesso à infra-estrutura pelos seus usuários, permitindo o acesso às suas funcionalidades somente por usuários previamente cadastrados na *Base de Dados de Usuários*. Além disso, este componente verifica o nível de permissão do usuário que efetuou *login*. Os níveis de permissão foram definidos de acordo com os papéis descritos no processo de testes descrito no Capítulo 4 (Gerente de Teste, Projetista de Teste e Testador). Após a validação do *login*, a infra-estrutura permitirá o acesso às funcionalidades dos componentes *Gerenciador da Equipe de Teste* e *Gerenciador de Testes Instanciados*.
- **Gerenciador da Equipe de Teste:** Este componente é responsável pelo cadastro na *Base de Dados de Usuários* dos membros que compõem a equipe de teste, que serão as pessoas que terão permissão de acesso à infra-estrutura e que poderão ser alocadas em processos de teste instanciados. Permite, também, a consulta aos dados dos membros cadastrados na infra-estrutura, possibilitando a visualização e impressão de um relatório com os dados de um ou vários membros da equipe de teste.
- **Gerenciador de Testes Instanciados:** Este componente é responsável pela instanciação de novos processos de teste a serem acompanhados utilizando a infra-estrutura e da consulta aos artefatos dos processos de teste já instanciados na *Base de Dados de Teste Instanciados*. Este componente também lista todas os processos de teste instanciados que ainda não foram concluídos nas quais o usuário que efetuou *login* na infra-estrutura está alocado, possibilitando o gerenciamento individualizado de cada processo de teste que esteja em andamento por meio do componente *Gerenciador de Processo de Teste*.
- **Gerenciador de Processo de Testes:** Este componente é responsável pelo gerenciamento do processo de testes a ser seguido para uma atividade de teste que esteja em andamento criada por meio da infra-estrutura. Este componente gerencia a execução das atividades a serem realizadas no processo de testes instanciado, de acordo com as dependências entre as atividades e do papel de

cada usuário dentro do processo de testes, possibilitando a visualização e impressão dos artefatos produzidos ao longo do processo. Permite, ainda, o controle do processo de testes a ser seguido por meio de consulta ao cronograma dos testes, informações sobre os membros da equipe de teste alocados para o processo de testes instanciado e gráficos de acompanhamento dos testes. Este componente é o responsável pela documentação e sistematização dos testes a partir da infra-estrutura.

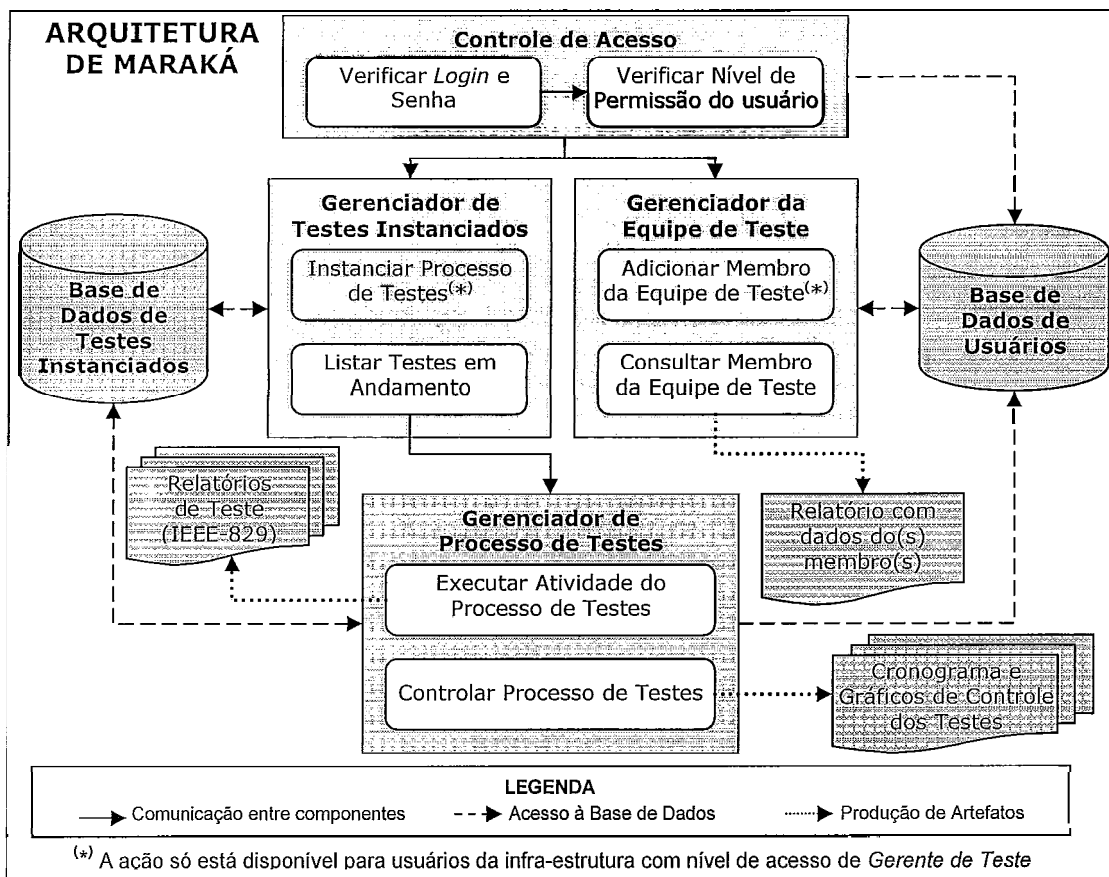


Figura 5.1. Arquitetura de Maraká

A partir da especificação da arquitetura da infra-estrutura e seus componentes, foram realizadas algumas decisões de projetos com o intuito de instanciá-la.

5.3.2 – Utilização do Framework Mambo

A partir da necessidade de utilização da infra-estrutura por meio da Internet (REQ 21 – seção 5.2.2) e controle do acesso dos usuários (REQ 22 – seção 5.2.2) foi cogitada a utilização de um *framework* para apoiar o atendimento a esses requisitos. Com isso, o

framework Mambo¹⁴ foi escolhido como base para o desenvolvimento da infra-estrutura proposta por possuir as seguintes características:

- É um gerente de conteúdo dinâmico em *sites* ou aplicações que utilizam a Internet como plataforma, gerenciando páginas, menus, *links*, usuários, nível de acesso, *template* (interface gráfica) da aplicação, idiomas e serviços específicos.
- Disponibiliza um mecanismo para controle de acesso aos usuários e ao conteúdo de uma aplicação.
- Simplifica a construção da interface gráfica de um *site* ou uma aplicação por meio de seus *templates*¹⁵.
- Permite a criação e instalação de novos componentes para a aplicação para a realização de novas funcionalidades que não são atendidas originalmente no *framework* (Figura 5.2 – *IMambo*).

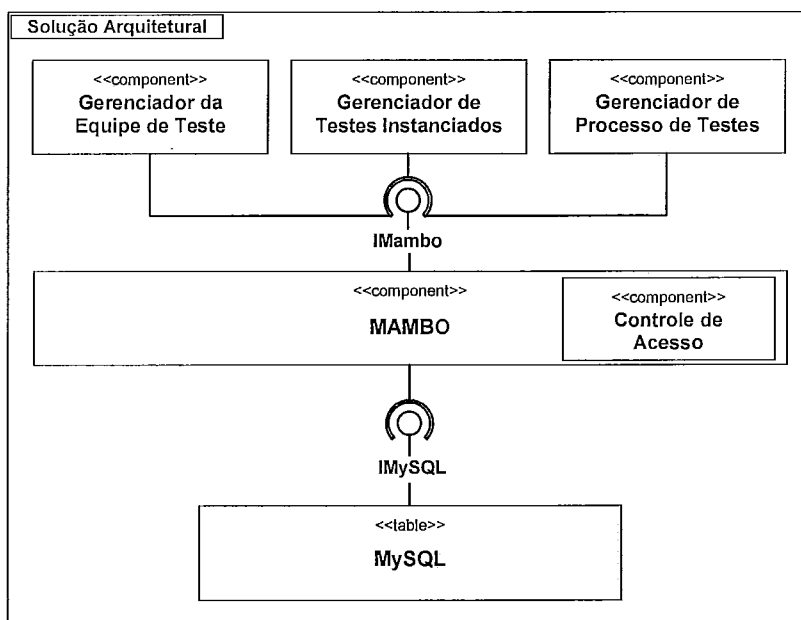


Figura 5.2. Camada criada pelo Mambo entre a aplicação e a base de dados

- Possui duas áreas para gerenciamento do conteúdo da aplicação: um *frontend*, que é a interface que disponibiliza as funcionalidades da aplicação aos seus usuários, e um *backend*, que é uma área destinada aos administradores da aplicação, onde novos componentes podem ser instalados, páginas podem ser criadas, menus ou *links* podem ser adicionados.

¹⁴ Mais informações sobre o *framework* Mambo podem ser obtidas em www.mamboforge.net.

¹⁵ *Template* é recurso do Mambo para definição da forma como os dados serão apresentados ao usuário (disposição dos elementos da aplicação nas páginas, layout gráfico, cabeçalhos, rodapés, formato e tamanho das fontes).

- Possui uma camada de interface com o banco de dados, de forma que os dados armazenados no banco de dados possam ser acessados por meio desta interface, reduzindo o esforço de inserção, alteração, remoção e consulta aos dados no banco de dados (Figura 5.2 – *IMySQL*).
- É um software que possui código aberto e possui diversos componentes disponíveis na sua comunidade (www.mamboforge.net) que podem ser instalados para acrescentar funcionalidades à aplicação.

Na arquitetura definida na seção 5.3.1, o *framework* Mambo instancia o componente *Controle de Acesso* e a *Base de dados de Usuários* (ver Figura 5.1). Dessa forma, foi necessário o desenvolvimento dos demais componentes e da *Base de Dados de Testes Instanciados*.

O *framework* Mambo é desenvolvido na plataforma de desenvolvimento PHP e utiliza o banco dados MySQL como repositório dos seus dados. Dessa forma, todos os componentes da infra-estrutura foram desenvolvidos nesta plataforma (PHP + MySQL).

5.3.3 – Diagrama de Classes

O passo seguinte na construção da infra-estrutura proposta foi a definição do seu projeto de alto nível, e para isso foi utilizado o diagrama de classes da UML (BOOCH *et al.*, 1998) como forma de representação.

Este diagrama contém as classes referentes ao Processo de Testes (suas atividades e dependências), suas instâncias a serem seguidas, aos membros da Equipe de Teste e aos Artefatos (e seus conteúdos) gerados a cada Processo de Teste Instanciado, conforme pode ser visualizado na Figura 5.3.

Com isso, a partir da definição dos requisitos a serem atendidos na infra-estrutura, da especificação de sua arquitetura, das decisões de projeto e do seu projeto de alto nível, o passo seguinte consiste na construção da infra-estrutura proposta. A seção seguinte apresentará a infra-estrutura Maraká por meio de suas principais funcionalidades.

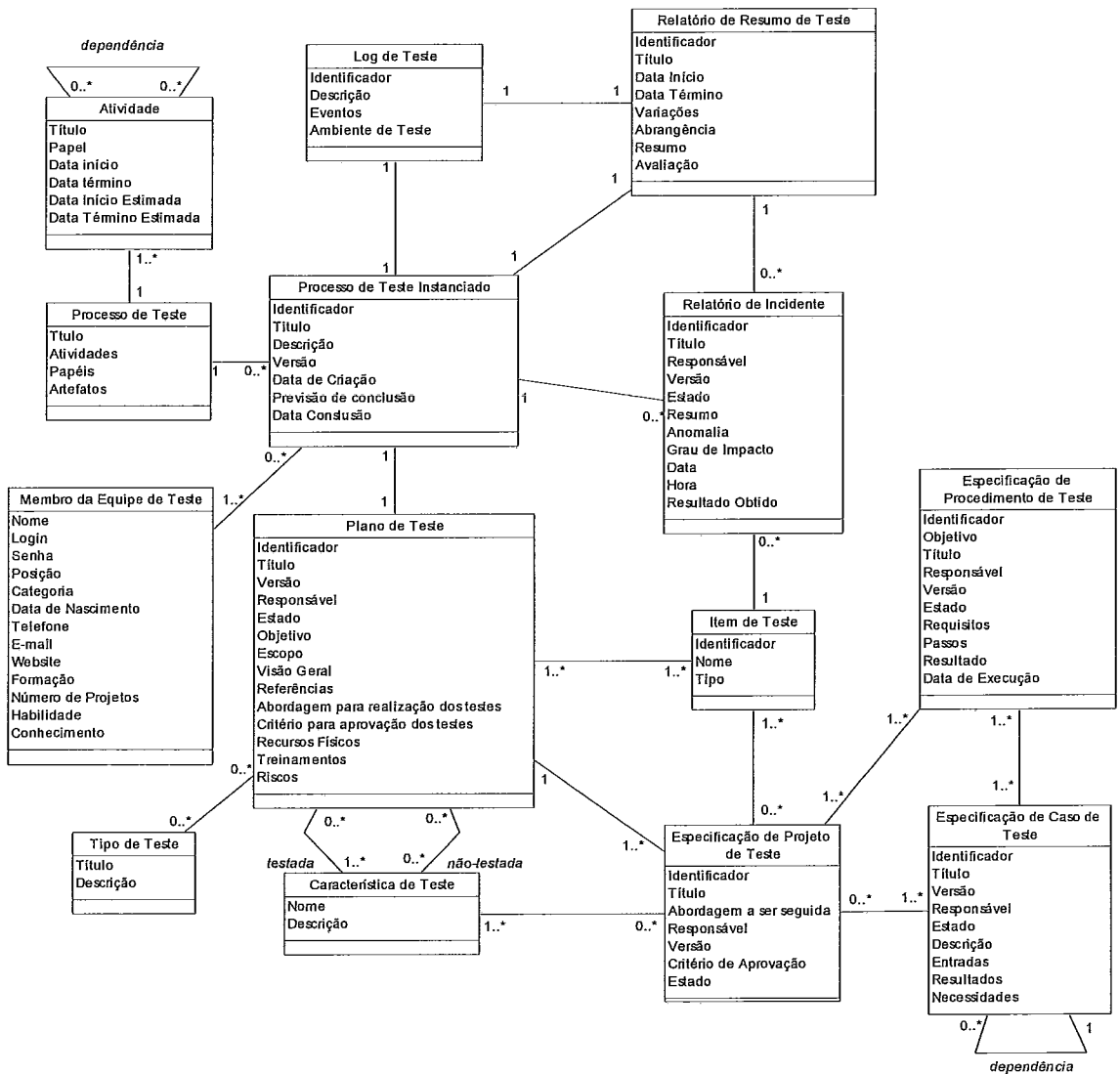


Figura 5.3. Diagrama de Classes de Maraká

5.4 – A Infra-estrutura Maraká

A seguir, estão descritas as funcionalidades de Maraká, considerando o perfil de um Gerente de Teste, pois este perfil possui acesso a todas as funcionalidades da infra-estrutura. Os demais perfis (Projetista de Teste e Testador) terão acesso a um sub-

conjunto dessas funcionalidades, de acordo com as suas atividades no processo de testes de software.

5.4.1 – *Login* e Menu Principal de Maraká

Conforme definido nos requisitos de Maraká, somente usuários cadastrados terão acesso à infra-estrutura. Desta forma, o primeiro passo na sua utilização é efetuar *login*. Em seguida, Maraká apresentará uma tela com as seguintes opções (Figura 5.4):

- **Testes em Andamento:** lista todos os processo de teste de software instanciados que estão em andamento nos quais o usuário que efetuou *login* em Maraká esteja alocado. A partir dessa opção, ele poderá selecionar um dos processos de testes instanciados para realizar a sua execução.
- **Equipe de Teste:** gerencia os dados dos membros da equipe de teste cadastrados na infra-estrutura. Permite que Gerentes de teste consultem as informações sobre os membros cadastrados e cadastrem novos membros da equipe de teste, que passarão a ter acesso à infra-estrutura.
- **Testes Instanciados:** gerencia os dados sobre os processos de teste instanciados para um projeto. Permite que qualquer usuário consulte os artefatos produzidos ao longo dos processos de teste instanciados. Além disso, permite que Gerentes de Teste instanciem novos processos de teste a serem acompanhados pela infra-estrutura.
- **Alterar Dados:** permite que qualquer usuário que efetuou *login* na infra-estrutura altere os dados da sua conta de acesso (e-mail e senha de acesso).
- **Configurações:** permite que Gerentes de Teste façam carga de dados que possam ser utilizados durante a execução do processo de testes de software, como: tipos de teste e características de teste. Permite, ainda, que somente Gerentes de Teste excluam Processos de Testes Instanciados por meio de Maraká.
- **Logout:** finaliza a execução de Maraká e retorna à página de *login*.

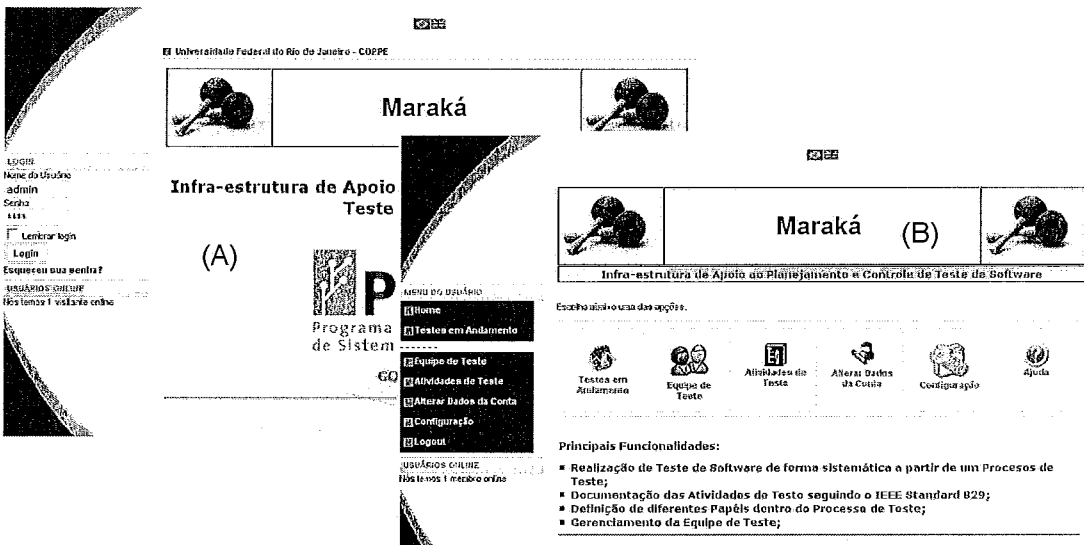


Figura 5.4. (A) Tela de *Login* e (B) Página Inicial de Maraká

5.4.2 – Gerenciador da Equipe de Teste

No gerenciamento da equipe de teste de software, Maraká permite que somente Gerentes de Teste cadastrem um novo membro da equipe de teste (ou alteração dos dados de um membro) (Figura 5.5). No momento do cadastro de um membro da equipe de teste, será enviada uma mensagem para o e-mail informado no cadastro confirmando seu *login* e senha, para que ele possa ter acesso à Maraká. Os membros cadastrados poderão ser alocados, posteriormente, em processos de testes instanciados por Maraká.

Novo Membro da Equipe de Teste.	
>>Nome:	Ariolo Claudio Dias Neto
>>Login:	acdn
>>Posição:	Gerente de Projeto
>>Categoria:	TestTeam/Gerente de Teste
>>Telefones:	(21) XXXX-6935
>>E-Mail:	acdn@cos.ufjf.br
>>Website:	www.cos.ufjf.br/~acdn
>>Data de Nascimento:	13/09/1982
>>Formação:	Ensino Superior
>>Total de Projetos que Participou:	4
>>Habilidades:	Trabalho em Equipe
>>Conhecimentos:	Técnicas de Teste Engenharia de Software Experimental CMMI e mg.s.BR



 **Salvar**  **Cancelar**







Figura 5.5. Cadastro ou alteração dos dados de um membro da equipe de teste

Maraká permite, ainda, que qualquer usuário consulte os dados dos membros cadastrados na equipe de teste (Figura 5.6).




Listar Todos

Consultar Membros da Equipe de Teste




Gerente de Teste

Nome	Posição	Telefone	E-Mail	Website	Testes que participou	Data de Nascimento
 Arilo Claudio Dias Neto	Gerente de Projeto	2542-6935			3	13/09/1982
 Dessana Redig	Analista	3654-3210			3	21/05/1984

Projetista de Teste

Nome	Posição	Telefone	E-Mail	Website	Testes que participou	Data de Nascimento
 José Santos Dias	Gerente de Projeto	2345-9372			14	06/12/1956

Testador

Nome	Posição	Telefone	E-Mail	Website	Testes que participou	Data de Nascimento
 Rafael Barcelos	Programador	3234-5678			1	28/03/1982




 Imprimir  Voltar

Figura 5.6. Consulta aos dados dos membros da equipe de teste

Além disso, permite que Gerentes de Teste visualizem/imprimam relatórios com os dados individuais de um membro ou de toda a equipe de teste cadastrada (Figura 5.7).



Relatório de Membro da Equipe de Teste

Gerado por Maraká - 11/02/2006

Nome: Arilo Claudio Dias Neto
Função no Processo de Teste: Projetista de Teste
Posição: Gerente de Projeto
Telefone: 2542-6935
E-mail: ariloclaudio@hotmail.com
Website: www.cos.ufrj.br/~acdn
Data de Nascimento: 13/09/1982
Número de Testes que Participou: 3
Habilidade: Liderança
Conhecimento: C++

Grupo de Engenharia de Software Experimental - COPPE/UFRRJ
<http://www.cos.ufrj.br/~ese>

Figura 5.7. Relatório com os dados de um membro da equipe de teste

5.4.3 – Gerenciador de Testes Instanciados

No gerenciamento de processos de teste de software instanciados, Maraká permite que somente Gerentes de Teste instanciem um processo de testes para um novo projeto (Figura 5.8).

Instanciando um Novo Processo de Testes	
>> Identificador	PT-001
>> Título	Teste de Unidade para o Sistema DIAS
>> Descrição	Este processo é responsável pelo planejamento e controle dos testes de unidade para o Sistema Dias, responsável pelo controle de estoque na empresa.
>> Versão	2.0
>> Responsável	Ariilo Claudio Dias Neto
>> Data de Criação	13/09/2005
>> Previsão de Encerramento	10/01/2006






Figura 5.8. Instanciação de um Processo de Testes de Software

Além disso, permite que qualquer membro da equipe de teste (usuário de Maraká) visualize/imprima os artefatos produzidos ao longo de um processo de testes instanciado pela infra-estrutura (Figura 5.9).



Processos de Testes de Software Instanciados					
Escolha um dos processos de testes listados e clique em seguida no botão CONSULTAR para visualizar os artefatos produzidos ao longo dos testes no processo escolhido.					
Parâmetros:					
Identificador:	<input type="text"/>	Título:	<input type="text"/>	<input type="button" value="Filtrar"/>	
Identificador	Título	Data de Criação	Previsão de Encerramento	Versão	
 PT-11	SIGDEM 2.0	02/09/2005	13/10/2005	1.0	
 TESTE01	Sistema DIAS LTDA	28/09/2005	31/12/1969	2.0	

Figura 5.9. Consulta aos artefatos de Processos de Teste Instanciados por Maraká

5.4.4 – Testes em Andamento

Maraká possui uma área específica onde estão listadas somente os processos de teste de software instanciados nos quais o usuário corrente está alocado e que ainda

estão em andamento, ou seja, não foram concluídos (Figura 5.10). Dessa forma, cada usuário poderá acompanhar a execução dos processos de testes instanciados nos quais está alocado, para a realização de suas atividades.

Testes em Andamento				
Escolha abaixo um dos processos de testes que está em andamento e que você está alocado. Após isso clique no botão ABRIR.				
Identificador	Título	Função	Estado	Versão
PT-11	SIGDEM 2.0	Gerente de Testes	Não Iniciado	1.0
TESTE01	Sistema DIAS LTDA	Gerente de Teste	Em Planejamento	1.2


Figura 5.10. Processos de Testes Instanciados que estão em andamento


5.4.5 – Gerenciador do Processo de Testes Instanciado


No gerenciamento de um processo de testes instanciado a partir de Maraká, a infra-estrutura permite a execução das macro-atividades e atividades do processo de testes descrito no Capítulo 4 e o controle dos testes que estão sendo realizados, de acordo com as opções apresentadas na Figura 5.11, que são:


Gerenciando o Processo de Teste – Projeto: Sistema DIAS LTDA


Escolha uma opção para acessar as informações sobre o Processo de Testes Instanciado.


 Processo de Teste de Software


 Cronograma dos Testes


 Artefatos do Processo de Testes Instanciado


 Visualizar Equipe de Teste


 Visualizar Gráficos dos Testes

Informações sobre o Processo de Testes Instanciado

>> Identificador:	TESTE01
>> Título:	Sistema DIAS LTDA
>> Descrição:	Sistema de Controle de Estoque para as organizações DIAS LTDA. Sistema responsável pelo cadastro dos itens do estoque, por efetuar baixa nos produtos e emitir um relatório com os itens abaixo do nível crítico definido.
>> Responsável:	Arilo Claudio Dias Neto
>> Versão:	1.2
>> Data de Criação:	28/09/2005
>> Previsão de Encerramento:	30/12/2005




 Editar
  Finalizar
  Voltar

Figura 5.11. Tela de Gerenciamento do Processo de Testes Instanciado

- **Processo de Testes de Software:**

Visualiza as macro-atividades e atividades do processo de testes de acordo com seu estado atual (Figura 5.12) (azul = concluída, verde = em andamento, azul ciano = não iniciada).

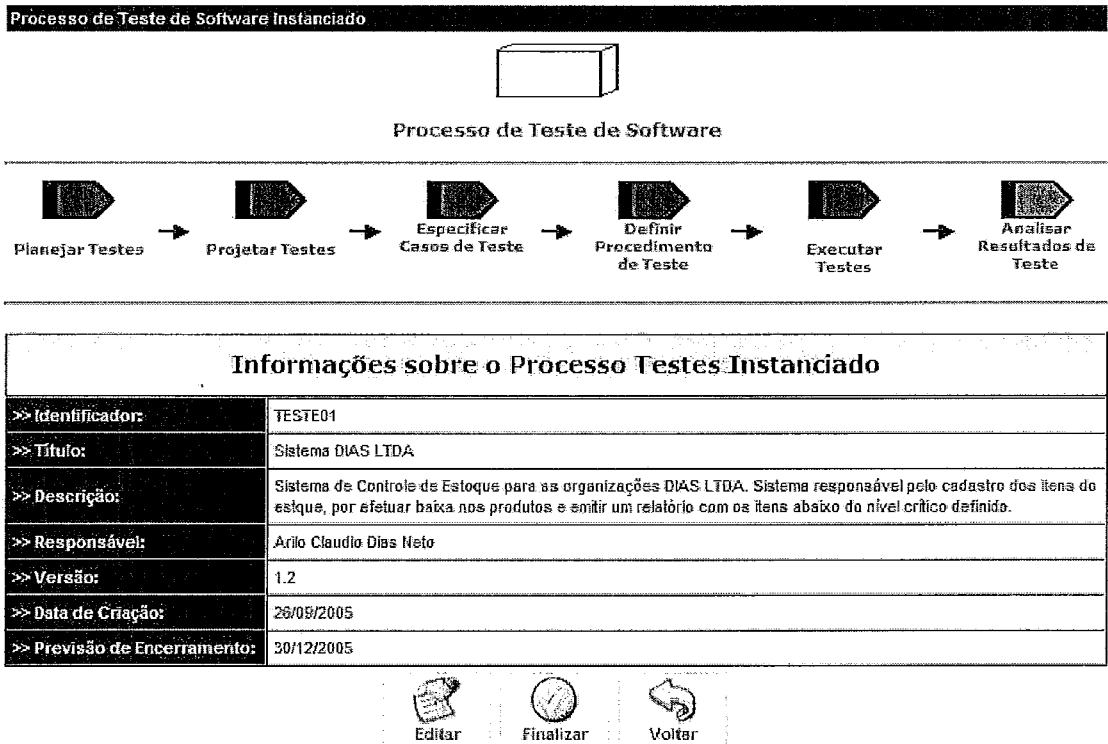


Figura 5.12. Processo de Testes de Software

Além disso, permite a execução das atividades de acordo com o papel do usuário no processo de testes. Em Maraká, o usuário poderá:

- Iniciar a execução das macro-atividades do processo de testes e suas atividades;
- Finalizar as macro-atividades e atividades do processo de testes;
- Criar artefatos, seguindo o processo de testes;
- Preencher os artefatos criados por meio das macro-atividades e atividades do processo de testes;
- Visualizar os artefatos criados em cada atividade do processo de testes (um exemplo de um Plano de Teste gerado pode ser visualizado na Figura 5.13);



Plano de Testes

Gerado por Maraká - 11/02/2006

Projeto: Sistema DIAS LTDA
Título: PLANO DE TESTE DE SISTEMA
Autor: Arilo Claudio Dias Neto
Data de Criação: 26/09/2005

Identificador do Plano: PLANO 01
Versão: 1.0
Status: Finalizado
Data de Conclusão: 18/10/2005

1. Introdução

1.1. Objetivo:

O objetivo dos testes é avaliar as funcionalidades definidas para o software.

1.2. Escopo:

Soamente testes funcionais serão realizados, nas seguintes funcionalidades: - Cadastro de Itens do Estoque; - Efetuar baixa no estoque; - Imprimir relatório de itens em nível crítico;

1.3. Visão Geral:

O sistema de controle de estoque é um dos módulos do sistema gerencial das organizações DIAS LTDA, e visa ao controle dos itens em estoque na organização.

Figura 5.13. Plano de Teste gerado por Maraká

Com o objetivo de facilitar a utilização de Maraká pelos seus usuários ao longo da execução de um processo de testes instanciado, foi definido um mecanismo de ajuda, de forma que em qualquer tela de uma das macro-atividades ou atividades do processo de testes o usuário poderá consultar uma descrição das informações a serem preenchidas por meio do botão *Ajuda* (em destaque na Figura 5.14). O texto de ajuda disponibiliza, ainda, um exemplo de preenchimento para a tela do processo (Figura 5.15).

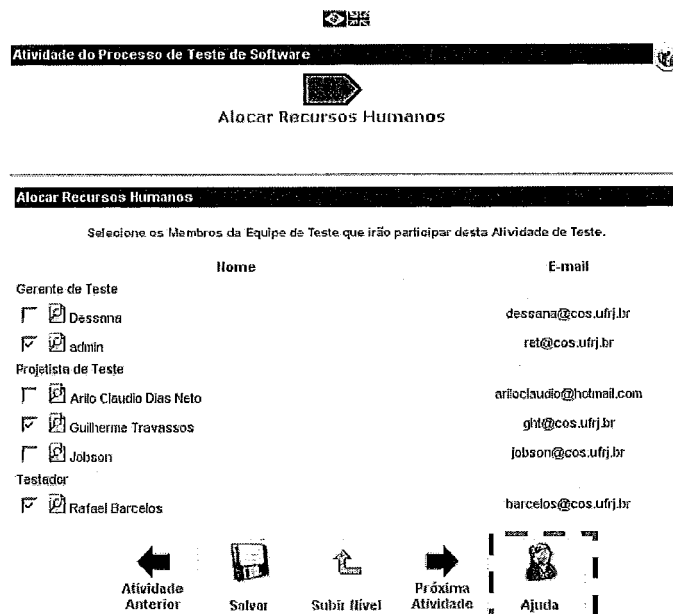


Figura 5.14. Página de uma atividade do Processo (Em destaque, botão *Ajuda*)



Maraká - Uma infra-estrutura Computacional de Apoio ao Planejamento e Controle de Teste de Software



Definir Itens de Teste:

- **ITENS DE TESTE:** Devem ser identificados os itens a serem testados, incluindo uma descrição de cada item e sua versão. A granularidade dos itens de teste dependem do projeto a ser testado, e podem ser: programas, módulos, procedimentos, outros (IEEE Standard 829).
- **TIPOS DE TESTE:** Devem ser selecionados os tipos dos testes a serem realizados. Os tipos de teste podem ser: unidade, integração, sistema, aceitação, regressão, outros.

A Figura abaixo mostra um exemplo desta tela preenchida.

Identificador	Título	Descrição	Versão
SCE-ADMIN	SCE DIAS LTDA - Módulo Administrador	(1)	1.0
SCE-VEENDA	SCE DIAS LTDA - Módulo de Vendas	(1)	1.1

Figura 5.15. Mecanismo de Ajuda à execução do Processo de Testes

• Cronograma do Processo de Testes Instanciado:

Visualiza o cronograma definido para os testes, confrontando as datas estimadas para as macro-atividades e atividades do processo e as datas reais, ou seja, as data em que essas macro-atividades e atividades estão sendo efetivamente realizadas (Figura 5.16). Maraká permite a impressão de um relatório com o cronograma dos testes.

Atividade	Pré-Atividade	Início Estimado	Término Estimado	Data de Início	Data de Término
Planejar Testes		17/10/2005	25/10/2005	18/10/2005	18/10/2005
Definir Objetivos/Esopo		17/10/2005	18/10/2005	18/10/2005	18/10/2005
Definir Itens de Teste	Definir Objetivos/Esopo	18/10/2005	18/10/2005	18/10/2005	18/10/2005
Definir Características de Teste	Definir Itens de Teste	18/10/2005	18/10/2005	18/10/2005	18/10/2005
Definir Critérios de Teste	Definir Características de Teste	19/10/2005	19/10/2005	18/10/2005	18/10/2005
Alocar Recursos Humanos	Definir Critérios de Teste	19/10/2005	19/10/2005	18/10/2005	18/10/2005
Alocar Recursos Fisicos	Alocar Recursos Humanos	20/10/2005	20/10/2005	18/10/2005	18/10/2005
Definir Cronograma	Alocar Recursos Fisicos	20/10/2005	21/10/2005	18/10/2005	18/10/2005
Identificar Riscos	Definir Cronograma	24/10/2005	25/10/2005	18/10/2005	18/10/2005
Projetar Testes	Planejar Testes			18/10/2005	20/10/2005
Caracterizar Projeto de Teste				18/10/2005	
Descrever Abordagem de Teste	Caracterizar Projeto de Teste			27/09/2005	
Identificar Casos de Teste	Descrever Abordagem de Teste			27/09/2005	
Identificar Procedimentos de Teste	Identificar Casos de Teste			27/09/2005	
Priorizar Procedimentos de Teste	Identificar Procedimentos de Teste			27/09/2005	

Figura 5.16. Cronograma dos Testes

- **Artefatos Produzidos ao longo dos Testes:**

Visualiza todos os artefatos produzidos ao longo do processo de testes instanciado, indicando o seu estado (em construção ou finalizado) (Figura 5.17). Maraká permite a impressão de cada artefato produzido ao longo dos testes.



















Visualização dos Artefatos de Teste	
PLANO DE TESTE	
 PT-SIGDEM - Plano de Teste de Sistema	 Finalizado
ESPECIFICAÇÕES DE PROJETO DE TESTE	
 TD-01 - Projeto de Teste - FUNCIONALIDADE	 Em Construção
 PRT-02 - Segundo Projeto de Teste	 Em Construção
 ProjetoTeste03 - Teste de Carga	 Finalizado
ESPECIFICAÇÕES DE CASO DE TESTE	
 CT-01 - Mensagem Transmitida - Fluxo Principal	 Em Construção
 CT-02 - Mensagem Transmitida - Fluxo Alternativo	 Finalizado
ESPECIFICAÇÕES DE PROCEDIMENTO DE TESTE	
 PT-01 - Mensagem Transmitida - Fluxo Principal	 Em Construção
 PT-02 - Mensagem Transmitida - Fluxo Alternativo 01	 Finalizado
 PT-03 - Mensagem Transmitida - Fluxo Alternativo 02	 Finalizado

Figura 5.17. Artefatos produzidos no Processo de Testes Instanciado

- **Visualizar Equipe de Teste:**

Visualiza a lista de membros da equipe de teste alocados para o processo de testes instanciado, indicando as atividades em que cada membro está alocado de acordo com o seu papel no processo de testes e os artefatos que produziram ao longo do processo (Figura 5.18). Maraká permite a impressão dessas informações para cada membro alocado para o processo de teste instanciado.

Visualizar Informações do Membro alocado no Processo de Testes

Dados Pessoais

Nome:	Rafael Barcelos
E-mail:	barcelos@cos.ufrj.br
Papel no Processo:	Testador

Atividades associadas a esse membro

Atividade	Pré-Atividade	Início Estimado	Término Estimado	Data de Início	Data de Término
Executar Testes	Definir Procedimento de Teste	31/12/1969	31/12/1969	24/10/2005	24/10/2005
Registrar Execução de Procedimento de Teste	Registrar Ambiente de Teste	31/12/1969	31/12/1969	17/10/2005	31/12/1969

Artefatos produzidos por este membro:

Especificação de Projeto de Teste

- TD-01 - Projeto de Teste - FUNCIONALIDADE

Especificação de Procedimento de Teste

- PT-02 - Mensagem Transmitida - Fluxo Alternativo 01






Figura 5.18. Informações de um membro alocado para o Processo de Teste Instanciado

- **Visualizar Gráficos dos Testes:**

Visualiza diversos gráficos que contribuem para o acompanhamento e controle dos testes que estão sendo realizados (Figura 5.19). Os gráficos disponibilizados por Maraká foram descritos no requisito 17 (REQ 17) na seção 5.2.1.2 deste capítulo.

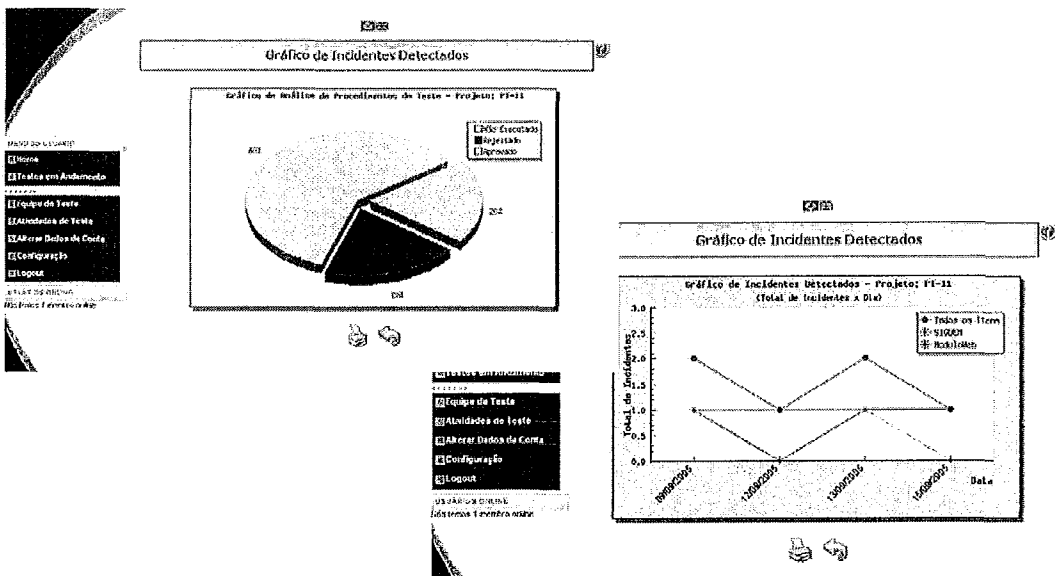
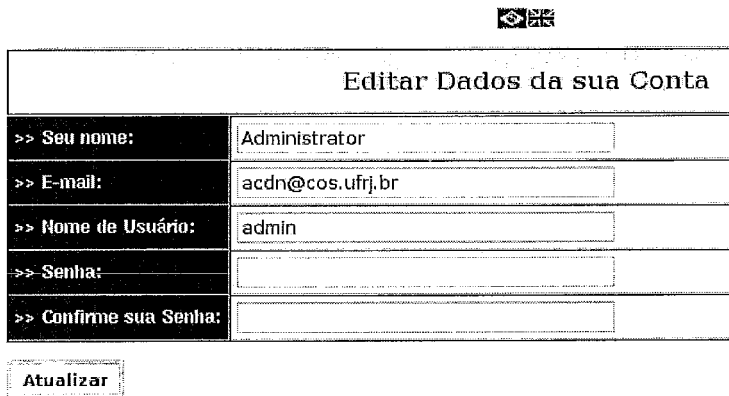


Figura 5.19. Gráficos de Controle dos Testes

5.4.6 – Alteração dos dados da conta de acesso à Maraká

Maraká permite que cada usuário altere informações de sua conta de acesso à infra-estrutura (e-mail e senha) (Figura 5.20).



Editar Dados da sua Conta	
>> Seu nome:	<input type="text" value="Administrator"/>
>> E-mail:	<input type="text" value="acdn@cos.ufrj.br"/>
>> Nome de Usuário:	<input type="text" value="admin"/>
>> Senha:	<input type="text"/>
>> Confirme sua Senha:	<input type="text"/>

Figura 5.20. Alteração dos dados da conta de acesso à Maraká

5.5 – Conclusão

Neste capítulo apresentamos uma proposta computacional de apoio ao planejamento e controle dos testes de software, seguindo as atividades definidas no processo de testes de software, permitindo a documentação e sistematização dessas atividades prevendo os diferentes papéis associados ao processo de testes de software.

Além disso, foram apresentados os requisitos funcionais e não-funcionais, as decisões de projeto, solução arquitetural e projeto de alto nível definido para a construção da infra-estrutura. Por fim, a infra-estrutura Maraká foi apresentada a partir de suas funcionalidades, como mecanismo computacional de apoio ao planejamento e controle dos testes de software. A infra-estrutura Maraká foi desenvolvida, desde a sua elaboração (definição de seus requisitos funcionais e não-funcionais) até os testes realizados, no período de Maio de 2005 à Dezembro de 2005 por somente uma pessoa (o autor deste trabalho) que atuava em média 4 horas por dia na sua construção.

O capítulo seguinte, "*Avaliação de Maraká em relação aos itens da Pesquisa de Opinião*", apresenta uma avaliação da infra-estrutura Maraká em relação ao atendimento às práticas de teste de software que compuseram a pesquisa de opinião apresentada no Capítulo 3.

Capítulo 6

Avaliação de Maraká em relação aos itens da Pesquisa de Opinião

Neste capítulo é descrita uma avaliação da infra-estrutura Maraká em relação ao apoio fornecido às práticas de teste de software que compuseram a pesquisa de opinião para caracterização da atividade de teste em ambientes reais de desenvolvimento, apresentada no Capítulo 3 deste trabalho.

6.1 – Introdução

Maraká foi desenvolvida visando a atender a necessidades reais de organizações de software, em relação às atividades de planejamento e controle de testes de software.

Com isso, neste Capítulo é apresentada uma avaliação da infra-estrutura Maraká em relação ao atendimento a cada um dos itens que foi avaliado na pesquisa de opinião descrita no Capítulo 3 deste trabalho¹⁶. O objetivo é avaliar o apoio fornecido por Maraká às práticas de teste de software questionadas na pesquisa de acordo com as necessidades reais de organizações, detectadas a partir da análise dos resultados obtidos.

Apesar dos resultados obtidos na pesquisa de opinião não poderem ser generalizados para um contexto mais amplo, eles apresentam indícios do que está sendo aplicado e das necessidades em relação às atividades de teste de software em organizações de software brasileiras localizadas em um cenário específico.

Sendo assim, a seção seguinte descreve a avaliação da infra-estrutura Maraká em relação a cada item avaliado na pesquisa de opinião. Por fim, na seção 6.3 são descritas as conclusões obtidas neste capítulo.

¹⁶ Os itens avaliados na pesquisa podem ser obtidos no questionário apresentado no Apêndice A.

6.2 – Avaliação de Maraká

Os itens analisados a seguir estão agrupados de acordo com a sua categoria (as categorias são: *Item Genérico, de Organização, de Planejamento, de Controle, de Medição e Análise e Ferramenta*), conforme descrito na seção 3.2.4 do Capítulo 3. Para cada item avaliado, serão apresentados o seu grau de importância e a avaliação de sua aplicabilidade, considerando o contexto geral das organizações que participaram do estudo, conforme os resultados descritos na seção 3.3.3 do Capítulo 3.

6.2.1 – Avaliação dos Itens Genéricos

- **Realização de Teste de Aceitação:**
 - Grau de Importância: 95,53%;
 - Aplicabilidade: *em 1 empresa não é aplicável; 1 empresa não utiliza; 4 empresas usam habitualmente; 3 empresas usam freqüentemente; 4 empresas usam como padrão em suas atividades;*
- **Realização de Teste de Unidade:**
 - Grau de Importância: 92,02%;
 - Aplicabilidade: *2 empresas não utilizam; 2 empresas usam habitualmente; 3 empresas usam freqüentemente; 6 empresas usam como padrão em suas atividades;*
- **Realização de Teste de Integração:**
 - Grau de Importância: 91,54%;
 - Aplicabilidade: *2 empresas usam habitualmente; 4 empresas usam freqüentemente; 7 empresas usam como padrão em suas atividades;*
- **Realização de Teste de Sistema:**
 - Grau de Importância: 90,98%;
 - Aplicabilidade: *1 empresa usa habitualmente; 4 empresas usam freqüentemente; 8 empresas usam como padrão em suas atividades;*
- **Realização de Teste de Regressão:**
 - Grau de Importância: 84,33%;
 - Aplicabilidade: *em 2 empresas não é aplicável; 2 empresas não utilizam; 4 empresas usam habitualmente; 2 empresas usam freqüentemente; 3 empresas usam como padrão em suas atividades;*

Maraká não foi especificada para apoiar um tipo de teste exclusivo, como teste de aceitação, unidade, integração, sistema, ou regressão. Dessa forma, a infra-estrutura pode ser utilizada para o planejamento e controle das atividades de teste independentemente do tipo de teste a ser realizado.

6.2.2 – Avaliação dos Itens de Organização

- Existência de um responsável ou uma equipe alocada para as atividades de teste:







- Grau de Importância: 95,98%;
- Aplicabilidade: 3 empresas não utilizam; 4 empresas usam freqüentemente; 6 empresas usam como padrão em suas atividades;

Maraká permite a especificação da equipe de teste de software para uma organização, de forma que somente essas pessoas definidas possam ser alocadas nos processos de testes (Figura 6.1).




Listar Todos

Consultar Membros da Equipe de Teste




Gerente de Teste

Nome	Posição	Telefone	E-Mail	Website	Testes que participou	Data de Nascimento
 Arilo Claudio Dias Neto	Gerente de Projeto	2542-8935			3	13/09/1982
 Dessana Redig	Analista	3654-3210			3	21/05/1984

Projetista de Teste

Nome	Posição	Telefone	E-Mail	Website	Testes que participou	Data de Nascimento
 José Santos Dias	Gerente de Projeto	2345-9372			14	06/12/1956

Testador

Nome	Posição	Telefone	E-Mail	Website	Testes que participou	Data de Nascimento
 Rafael Barcelos	Programador	3234-5678			1	28/03/1982



 Imprimir  Voltar

Figura 6.1. Gerenciamento da equipe de teste de software

No entanto, não pode ser garantida a existência de uma equipe ou responsável pelos testes, pois isso depende da estrutura da organização e não faz parte do escopo de Maraká.

- **Re-execução dos testes quando o software é modificado:**

- Grau de Importância: 93,30%;
- Aplicabilidade: 1 empresa não utiliza; 5 empresas usam habitualmente; 1 empresa usa freqüentemente; 6 empresas usam como padrão em suas atividades;

Maraká não possui atualmente um mecanismo para re-execução dos testes planejados. Atualmente, os testes executados ficam registrados na infra-estrutura, mas para re-executar os testes e registrar os novos resultados obtidos, é necessário criar uma nova instância do processo de testes e segui-lo novamente. No entanto, para facilitar, Maraká armazena os casos e procedimentos de teste utilizados durante a execução dos testes.

- **Utilização de metodologia ou processo para sistematizar e organizar testes de software:**

- Grau de Importância: 90,36%;
- Aplicabilidade: 2 empresas não utilizam; 2 empresas usam habitualmente; 4 empresas usam freqüentemente; 5 empresas usam como padrão em suas atividades;

Maraká possui um processo de testes de software inserido na infra-estrutura, de forma que todos os testes a serem realizados devem seguir este processo, considerando-se as dependências entre as atividades, os papéis definidos no processo e os artefatos a serem produzidos (Figura 6.2). O processo de testes de software foi definido no Capítulo 4 deste trabalho, mas precisa ser adaptado a uma organização para sua utilização em projetos de software na indústria.

- **Separação das atividades de teste do desenvolvimento:**

- Grau de Importância: 89,31%;
- Aplicabilidade: 3 empresas não utilizam; 2 empresas usam habitualmente; 5 empresas usam freqüentemente; 3 empresas usam como padrão em suas atividades;

Maraká segue um processo de testes de software composto por atividades e papéis independente do processo de desenvolvimento para a realização dos testes, havendo pontos para interação entre esses processos. No entanto, não pode ser garantida a utilização de Maraká de forma separada ao desenvolvimento, pois isso depende da organização que esteja utilizando a infra-estrutura e não faz parte do escopo de Maraká.



Figura 6.2. Em destaque, Processo de Testes de Software seguido por Maraká

• **Realização de treinamentos sobre testes:**

- Grau de Importância: 85,69%;
- Aplicabilidade: 5 empresas não utilizam; 5 empresas usam habitualmente; 1 empresa usa freqüentemente; 2 empresas usam como padrão em suas atividades;

Maraká permite somente que durante a atividade de planejamento dos testes, os treinamentos necessários para a realização dos testes para aquele projeto sejam especificados, indicando a habilidade necessária para a realização dos testes, a carga horária do treinamento e a previsão de dias a serem gastos com o treinamento (Figura 6.3). No entanto, não pode ser garantida a realização do treinamento ou a sua qualidade, pois isso não faz parte do escopo de Maraká.

Novo Treinamento	
>> Título	<input type="text" value="Treinamento de Particionamento por Equivalência"/>
>> Descrição	<input type="text" value="Realizar treinamento com membros da equipe de teste do projeto para utilização da técnica de teste de particionamento por equivalência."/>
>> Habilidade Necessária	<input type="text" value="Conhecimento básico sobre testes de software e desenvolvimento de software."/>
>> Carga Horária	<input type="text" value="40 horas"/>
>> Previsão	<input type="text" value="1 semana"/>






Figura 6.3. Registrando um treinamento de teste de software em Maraká

- **Existência de testadores em tempo integral para realização de testes:**

- Grau de Importância: 84,98%;
- Aplicabilidade: *em 1 empresa não é aplicável; 4 empresas não utilizam; 2 empresas usam habitualmente; 1 empresa usa freqüentemente; 5 empresas usam como padrão em suas atividades;*

Maraká permite a especificação da equipe de teste para uma organização, de forma que somente essas pessoas definidas possam ser alocadas nos processos de testes instanciados. No entanto, não pode ser garantida a utilização dessas pessoas nos testes em tempo integral, pois isso depende da estrutura da organização e não faz parte do escopo de Maraká.

6.2.3 – Avaliação dos Itens de Planejamento

- **Documentação do Plano de Teste descrevendo objetivos/abordagens:**

- Grau de Importância: 95,17%;
- Aplicabilidade: *1 empresa não utiliza; 1 empresa usa habitualmente; 4 empresas usam freqüentemente; 7 empresas usam como padrão em suas atividades;*

Maraká permite a criação do documento *Plano de Teste* (Figura 6.4) de acordo com o IEEE Standard 829 (1998), descrito na seção 4.3 deste trabalho. Neste documento está incluída uma seção com a definição dos objetivos e escopo dos testes a serem realizados.


	<h2>Plano de Testes</h2> <p><i>Gerado por Maraká - 11/02/2006</i></p>
<hr/>	
<p>Projeto: Sistema DIAS LTDA Identificador do Plano: PLANO 01 Título: PLANO DE TESTE DE SISTEMA Versão: 1.0 Autor: Arilo Claudio Dias Neto Status: Finalizado Data de Criação: 26/09/2005 Data de Conclusão: 18/10/2005</p>	
<hr/>	
<p>1. Introdução</p>	
<p>1.1. Objetivo: O objetivo dos testes é avaliar as funcionalidades definidas para o software.</p>	
<p>1.2. Escopo: Somente testes funcionais serão realizados, nas seguintes funcionalidades: - Cadastro de Itens do Estoque; - Efetuar baixa no estoque; - Imprimir relatório de itens em nível crítico;</p>	
<p>1.3. Visão Geral: O sistema de controle de estoque é um dos módulos do sistema gerencial das organizações DIAS LTDA, e visa ao controle dos itens em estoque na organização.</p>	

Figura 6.4. Plano de Teste gerado por Maraká

• **Documentação de Procedimentos e Casos de Teste e seus Resultados:**

- Grau de Importância: 89,67%;
- Aplicabilidade: 1 empresa não utiliza; 2 empresas usam habitualmente; 3 empresas usam freqüentemente; 7 empresas usam como padrão em suas atividades;

Maraká permite a criação dos documentos *Especificação de Caso de Teste* e *Especificação de Procedimento de Teste* para cada caso ou procedimento de teste, respectivamente (Figura 6.5), de acordo com o IEEE Standard 829 (1998). Da mesma forma, após a execução dos testes, os resultados dos procedimentos de teste executados são registrados no documento *Histórico dos Testes*, de acordo com o IEEE Standard 829 (1998). Estes documentos são criados seguindo o processo de teste definido em Maraká.



Especificação de Caso de Teste
Gerado por Maraká - 11/02/2006


Projeto: TESTE01 - Sistema DIAS LTDA
Identificador do Caso de Teste: CTE-01
Título: Cadastro de Item falhou **Versão:** 1.0
Autor: Guilherme Travassos **Status:** Finalizado
Data de Criação: 21/09/2005 **Data de Conclusão:** 23/09/2005

1. Caracterização do Caso de Teste

1.1. Descrição do Caso de Teste:
Colocar um identificador do item inválido. Dessa forma, o sistema deverá apresentar uma mensagem de erro.

1.2. Itens de Teste:

- Item de Teste: SCE-ADMIN - SCE DIAS LTDA - Módulo Administrador



Especificação de Procedimento de Teste
Gerado por Maraká - 11/02/2006

Projeto: TESTE01 - Sistema DIAS LTDA
Identificador: SPT-01
Título: Falhando o Cadastro de Item **Versão:** 1.0
Autor: Guilherme Travassos **Status:** Em andamento
Data de Criação: 12/10/2005 **Data de Conclusão:** 17/10/2005

1. Objetivo do Procedimento de Teste
Avaliar se o sistema apresenta a mensagem de erro na tentativa de cadastrar um item com dados incorretos.

2. Requisitos do Procedimento de Teste
Computador com o sistema Operacional Windows XP.

Figura 6.5. Documentação de Caso e Procedimento de Teste gerado por Maraká

• **Identificação e utilização de riscos para projetar, organizar e executar testes de software:**

- Grau de Importância: 77,63%;
- Aplicabilidade: *em 1 empresa não é aplicável; 4 empresas não utilizam; 6 empresas usam habitualmente; 6 empresas usam freqüentemente em suas atividades;*

Maraká permite, durante o planejamento dos testes, a especificação dos riscos associados à atividade de teste. Devem ser analisados a probabilidade de ocorrência e o grau de impacto de cada risco, e devem ser especificados um plano de mitigação e contingência para eles (Figura 6.6).

• **Desenvolvimento de testes de software antes da codificação:**

- Grau de Importância: 74,48%;
- Aplicabilidade: *6 empresas não utilizam; 5 empresas usam habitualmente; 2 empresas usam freqüentemente em suas atividades;*

Maraká possui um processo de testes a ser seguido, que é composto por atividades de planejamento e execução dos testes. As atividades de planejamento permitem que os testes sejam iniciados antes da codificação do software por meio da análise dos documentos do projeto, como documento de especificação de requisitos ou diagramas de projeto de alto nível, para a elaboração do plano, casos e procedimentos de

teste. No entanto, não pode ser garantido o início dos testes antes da codificação, pois isso depende da organização que esteja utilizando a infra-estrutura, e não faz parte do escopo de Maraká.

Novo Risco de Teste	
>> Nome	<input type="text"/>
>> Descrição	<input type="text"/>
>> Probabilidade de Ocorrência	Baixa - 0.0 a 0.3
>> Grau de Impacto	Baixa - 0.0 a 0.3
>> Plano de Contingência	<input type="text"/>
>> Plano de Mitigação	<input type="text"/>



 Salvar
  Cancelar

Figura 6.6. Registro de um risco associado à Atividade de Teste

6.2.4 – Avaliação dos Itens de Controle

- Registro de falhas e defeitos detectados durante os testes:

- Grau de Importância: 93,85%;
- Aplicabilidade: 1 empresa não utiliza; 5 empresas usam freqüentemente; 7 empresas usam como padrão em suas atividades;

Maraká permite o registro dos incidentes detectados durante os testes (ver Figura 6.7), criando para cada incidente, um *Relatório de Incidente de Teste* de acordo com o IEEE Standard 829 (1998).

Editar Incidente de Teste	
>> Identificador	Inc01
>> Data do Incidente	2005-09-09
>> Hora em que ocorreu	12:30
>> Resumo do Incidente	<input type="text"/>
>> Procedimento de Teste Associado	PT-02
>> Passo do Procedimento de Teste	passo 1
>> Entradas	<input type="text"/>
>> Resultados Esperados	<input type="text"/>

Figura 6.7. Registro dos Incidentes de Teste em Maraká

- **Conservação dos dados dos testes para usos futuros:**

- Grau de Importância: 84,87%;
- Aplicabilidade: 3 empresas não utilizam; 2 empresas usam habitualmente; 4 empresas usam freqüentemente; 4 empresas usam como padrão em suas atividades;

Maraká armazena todos os dados gerados ao longo dos processos de testes instanciados e permite a qualquer momento a consulta a esses dados, como um mecanismo de apoio durante a realização de novas atividades de teste (Figura 6.8).

Processos de Testes de Software Instanciados

Escolha um dos processos de testes listados e clique em seguida no botão CONSULTAR para visualizar os artefatos produzidos ao longo dos testes no processo escolhido.

Parâmetros:

Identificador: Título:



Identificador	Título	Data de Criação	Previsão de Encerramento	Versão
 PT-11	SIGDEM 2.0	02/09/2005	13/10/2005	1.0
 TESTE01	Sistema DIAS LTDA	26/09/2005	31/12/1999	2.0

Figura 6.8. Consulta aos dados dos processos de teste instanciados por Maraká

- **Avaliação dos documentos de teste (testware) quanto à sua qualidade:**

- Grau de Importância: 84,44%;
- Aplicabilidade: em 1 empresa não é aplicável; 3 empresas não utilizam; 2 empresas usam habitualmente; 5 empresas usam freqüentemente; 2 empresas usam como padrão em suas atividades;

Maraká permite somente a criação dos documentos de teste, mas não possui mecanismo de apoio à avaliação da qualidade desses documentos. Essa tarefa não faz parte do escopo de Maraká.

- **Registro do tempo gasto em testes:**

- Grau de Importância: 82,95%;
- Aplicabilidade: em 2 empresas não é aplicável; 3 empresas não utilizam; 4 empresas usam habitualmente; 2 empresas usam freqüentemente; 2 empresas usam como padrão em suas atividades;

Maraká registra automaticamente as datas de início e conclusão de cada atividade do processo de testes. As datas ficam registradas na infra-estrutura e podem ser visualizadas por meio do cronograma dos testes. Este cronograma pode ser consultado ou impresso pelos membros da equipe de teste a qualquer instante (Figura 6.9).

Cronograma dos Testes					
Lista das atividades que compõem o processo de testes. Estão especificadas as atividades precursora e as datas estimada e real de início e término para cada atividade.					
Atividade	Pré-Atividade	Início Estimado	Término Estimado	Data de Início	Data de Término
Planejar Testes:		17/10/2005	25/10/2005	18/10/2005	18/10/2005
↳ Definir Objetivos/Esopo		17/10/2005	18/10/2005	18/10/2005	18/10/2005
↳ Definir Itens de Teste	Definir Objetivos/Esopo	18/10/2005	16/10/2005	18/10/2005	18/10/2005
↳ Definir Características de Teste	Definir Itens de Teste	18/10/2005	18/10/2005	18/10/2005	18/10/2005
↳ Definir Critérios de Teste	Definir Características de Teste	19/10/2005	19/10/2005	18/10/2005	18/10/2005
↳ Alocar Recursos Humanos	Definir Critérios de Teste	19/10/2005	19/10/2005	18/10/2005	18/10/2005
↳ Alocar Recursos Físicos	Alocar Recursos Humanos	20/10/2005	20/10/2005	18/10/2005	18/10/2005
↳ Definir Cronograma	Alocar Recursos Físicos	20/10/2005	21/10/2005	18/10/2005	18/10/2005
↳ Identificar Riscos	Definir Cronograma	24/10/2005	25/10/2005	18/10/2005	18/10/2005
Projetar Testes:	Planejar Testes			18/10/2005	20/10/2005
↳ Caracterizar Projeto de Teste				18/10/2005	
↳ Descrever Abordagem de Teste	Caracterizar Projeto de Teste			27/09/2005	
↳ Identificar Casos de Teste	Descrever Abordagem de Teste			27/09/2005	
↳ Identificar Procedimentos de Teste	Identificar Casos de Teste			27/09/2005	
↳ Priorizar Procedimentos de Teste	Identificar Procedimentos de Teste			27/09/2005	

Figura 6.9. Cronograma dos Testes controlado por Maraká


- **Monitoração da aderência ao processo de teste de software:**
 - Grau de Importância: 79,56%;
 - Aplicabilidade: 6 empresas não utilizam; 3 empresas usam habitualmente; 2 empresas usam freqüentemente; 2 empresas usam como padrão em suas atividades;

Maraká possui diversos mecanismos para monitoração e acompanhamento do processo de teste, tais como:

- a comparação entre os cronogramas estimados e o cronograma real para os testes;
- o acompanhamento da execução das atividades definidas para cada membro da equipe alocado para o processo de testes;
- a visualização gráfica das atividades do processo de testes por meio de um gráfico de *gantt*;

- a visualização de um gráfico indicando o status (não executado, aprovado ou rejeitado) dos procedimentos de teste elaborados;
 - a visualização dos incidentes ocorridos em cada item de teste em duas visões: por data de ocorrência ou por item de teste.
- **Conservação de registro (log) dos testes executados:**
 - Grau de Importância: 78,91%;
 - Aplicabilidade: 3 empresas não utilizam; 1 empresa usa habitualmente; 4 empresas usam freqüentemente; 5 empresas usam como padrão em suas atividades;

Maraká permite o registro e armazenamento dos eventos ocorridos durante os testes, permitindo a consulta a esses dados em qualquer momento pelos seus membros da equipe de teste. Esses dados ficam armazenados no documento *Histórico dos Testes* (Figura 6.10).



Histórico dos Testes

Gerado por Maraká - 11/02/2006

Projeto: TESTE01 - Sistema DIAS LTDA	Identificador: LOG-01
Data de Início: 17/10/2005	Data de Término: 17/10/2005

1. Descrição
Execução dos Procedimentos de teste definidos.

2. Entradas de Atividades e Eventos
17/10/2005

- Evento: Execução - Falhando o Cadastro de Item
- Descrição: Procedimento que tenta cadastrar um item do estoque com dados incorretos.
- Status: Rejeitado
- Resultados Obtidos: O Sistema aceitou o cadastro do item, apesar dos dados incorretos.
- Incidentes de Teste: RIT-01
 - Impacto: Médio - 0.4 a 0.7
 - Item: SCE-ADMIN

Grupo de Engenharia de Software Experimental - COPPE/UFRJ
<http://www.cos.ufrj.br/~esc>

Figura 6.10. Histórico dos Testes gerado por Maraká

6.2.5 – Avaliação dos Itens de Medição e Análise

- **Análise e medição da cobertura dos testes:**

- Grau de Importância: 84,36%;
- Aplicabilidade: 6 empresas não utilizam; 2 empresas usam habitualmente; 2 empresas usam freqüentemente; 3 empresas usam como padrão em suas atividades;

Essa atividade não faz parte do escopo de Maraká. Os casos e procedimentos devem ser identificados de forma externa à infra-estrutura, de forma que a cobertura dos testes não seja controlada por ela. Maraká permite somente a especificação dos casos e procedimentos de teste identificados, gerando ao final a sua documentação.

- **Análise de defeitos identificados durante os testes para descobrir suas causas e como foram introduzidos (rastreadabilidade):**

- Grau de Importância: 82,92%;
- Aplicabilidade: 4 empresas não utilizam; 3 empresas usam habitualmente; 3 empresas usam freqüentemente; 3 empresas usam como padrão em suas atividades;

Maraká permite o registro dos incidentes ocorridos durante os testes e disponibiliza um gráfico indicando a quantidade de incidentes registrados para cada item testado, como forma de contribuir para a rastreabilidade das falhas e execução do processo de depuração no item que originou cada incidente (Figura 6.11). No entanto, ela não possui um mecanismo automático de apoio à análise dos defeitos identificados.

- **Medição dos testes e rastreamento dos custos com os testes:**

- Grau de Importância: 75,63%;
- Aplicabilidade: em 2 empresas não é aplicável; 5 empresas não utilizam; 5 empresas usam habitualmente; 1 empresa usa freqüentemente em suas atividades;

Maraká permite a especificação dos recursos necessários para a realização dos testes e seus valores. No entanto, não possui um mecanismo para medição dos testes e rastreamento dos custos associados a um processo de testes instanciado.

Gráfico de Incidentes Detectados

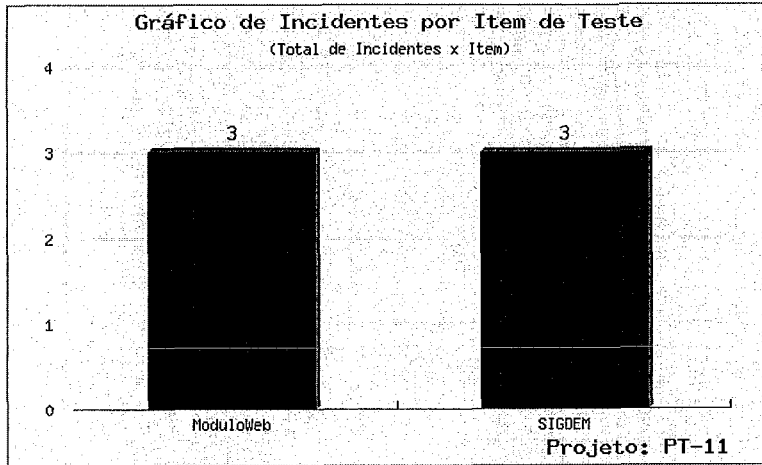


Figura 6.11. Gráfico de Incidentes registrados por Item de testado

- **Análise regular do padrão (tendência) de falhas e defeitos encontrados:**
 - Grau de Importância: 74,75%;
 - Aplicabilidade: em 1 empresa não é aplicável; 4 empresas não utilizam; 4 empresas usam habitualmente; 3 empresas usam freqüentemente; 1 empresa usa como padrão em suas atividades;
- **Medição da densidade de defeito (defeitos por mil linhas de código)**
 - Grau de Importância: 61,66%;
 - Aplicabilidade: em 1 empresa não é aplicável; 8 empresas não utilizam; 2 empresas usam habitualmente; 1 empresa usa freqüentemente; 1 empresa usa como padrão em suas atividades;

Maraká permite o registro dos incidentes ocorridos durante os testes e apresenta dois gráficos que apóiam a sua visualização: gráfico de incidentes ocorridos por data e incidentes ocorridos por item de teste. No entanto, não possui um mecanismo para análise do padrão de falhas detectadas e nem de medição da densidade de defeitos. Essas duas atividades devem ser realizadas externamente à Maraká.

6.2.6 – Avaliação dos Itens de Ferramenta

- **Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste:**
 - Grau de Importância: 84,43%;
 - Aplicabilidade: 7 empresas não utilizam; 3 empresas usam habitualmente; 1 empresa usa freqüentemente; 2 empresas usam como padrão em suas atividades;

Maraká é uma ferramenta de apoio à gerência dos testes, e permite, entre outras tarefas, o registro dos resultados obtidos na execução dos testes.

- **Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos:**
 - Grau de Importância: 80,05%;
 - Aplicabilidade: 7 empresas não utilizam; 3 empresas usam habitualmente; 1 empresa usa freqüentemente; 2 empresas usam como padrão em suas atividades;

Maraká é uma ferramenta de apoio à gerência dos testes, e permite o acompanhamento às atividades e artefatos do processo de teste, indicando seu status (em andamento ou finalizado), as datas de criação e conclusão e uma visualização gráfica da execução das atividades do processo por meio de um gráfico de *ganttt* (Figura 6.12).

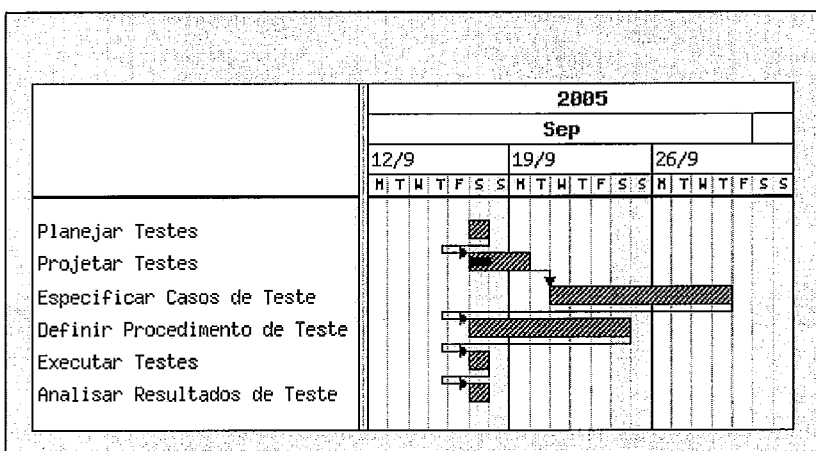


Figura 6.12. Gráfico de Gantt para um Processo de Testes Instanciado

- **Utilização de ferramentas para estimar esforço de teste e/ou cronograma:**

- Grau de Importância: 79,67%;
- Aplicabilidade: *8 empresas não utilizam; 3 empresas usam habitualmente; 1 empresa usa frequentemente; 1 empresa usa como padrão em suas atividades;*

Maraká permite a descrição das datas estimadas para cada atividade do processo de teste e permite a alocação de pessoas para as atividades de teste. No entanto, não possui um mecanismo de apoio a geração automática dessas estimativas. Desta forma, a estimativa das datas e a definição da alocação de recursos humanos devem ser definidas pelos usuários da infra-estrutura, a partir de seu conhecimento ou do uso de ferramentas externas à infra-estrutura.

- **Utilização de ferramentas geradoras de procedimentos ou casos de testes:**

- Grau de Importância: 74,95%;
- Aplicabilidade: *em 1 empresa não é aplicável; 6 empresas não utilizam; 5 empresas usam habitualmente; 1 empresa usa como padrão em suas atividades;*

Maraká permite somente a especificação dos casos e procedimentos de teste. A geração desses elementos não faz parte do seu escopo. Dessa forma, o usuário deve gerar os casos e procedimentos de teste manualmente ou a partir da utilização de uma ferramenta externa à infra-estrutura.

- **Disponibilidade de um banco de dados de teste (pacote de testes que simulam o ambiente de teste) para reutilização:**

- Grau de Importância: 82,69%;
- Aplicabilidade: *5 empresas não utilizam; 5 empresas usam habitualmente; 3 empresas usam como padrão em suas atividades;*

- **Utilização de ferramentas de medida de cobertura nas fases de teste (unidade, integração, sistema):**

- Grau de Importância: 78,39%;
- Aplicabilidade: *6 empresas não utilizam; 4 empresas usam habitualmente; 3 empresas usam como padrão em suas atividades;*

O propósito de Maraká é apoiar o planejamento e controle dos testes por meio da sistematização e documentação dos testes. Desta forma, esta prática não é atendida pela infra-estrutura.

6.3 – Conclusão

Este Capítulo apresentou uma avaliação da infra-estrutura Maraká, onde se analisou o atendimento às práticas de teste de software que compuseram a pesquisa de opinião descrita no Capítulo 3. Desta forma, foi descrito o apoio de Maraká a partir das necessidades reais de organizações, analisados a partir dos resultados da pesquisa.

A Tabela 6.1 apresenta o resumo da avaliação de prática de teste avaliada, possibilitando uma melhor visualização de quais práticas são atendidas pela infra-estrutura e qual o nível de atendimento a elas. Os itens estão agrupados de acordo com a sua categoria: *Genérico, Organização, Planejamento, Controle, Medição e Análise e Ferramenta*. Nesta tabela, o ícone ✓ representa o atendimento total de Maraká ao item em questão, o ícone ⚠ representa o atendimento parcial de Maraká ao item e o ícone ✗ indica que o item não foi atendido pela infra-estrutura, conforme descrito detalhadamente na seção 6.2 deste Capítulo.

Tabela 6.1 – Atendimento de Maraká às práticas de teste de software que compuseram a pesquisa de opinião.

ITENS DO QUESTIONÁRIO		ATENDIMENTO
Genérico	Realização de Teste de Aceitação	✓
	Realização de Teste de Unidade	✓
	Realização de Teste de Integração	✓
	Realização de Teste de Sistema	✓
	Realização de Teste de Regressão	✓
Organização	Existência de um responsável ou equipe alocada para as atividades de teste	✓
	Re-execução dos testes quando o software é modificado	✗
	Utilização de metodologia ou processo para sistematizar e organizar os testes	✓
	Separação das atividades de teste do desenvolvimento	✓
	Realização de treinamentos sobre testes	⚠
	Existência de testadores em tempo integral para realização de testes	⚠

Planejamento	Documentação do Plano de Teste descrevendo objetivos/abordagens	✓
	Documentação de Procedimentos e Casos de Teste e seus Resultados	✓
	Identificação e utilização de riscos para projetar, organizar e executar testes de software	✓
	Desenvolvimento de testes de software antes da codificação	⚠
Controle	Registro de falhas e defeitos detectados durante os testes	✓
	Conservação dos dados dos testes para usos futuros	✓
	Avaliação dos documentos de teste (<i>testware</i>) quanto à sua qualidade	✗
	Registro do tempo gasto em testes	✓
	Monitoração da aderência ao processo de teste de software	✓
	Conservação de registro (<i>log</i>) dos testes executados	✓
Medição e Análise	Análise e medição da cobertura dos testes	✗
	Análise de defeitos identificados durante os testes para descobrir suas causas e como foram introduzidos (<i>rastreabilidade</i>)	⚠
	Medição dos testes e rastreamento dos custos com os testes	⚠
	Análise regular do padrão (<i>tendência</i>) de falhas e defeitos encontrados	⚠
	Medição da densidade de defeito (defeitos por mil linhas de código)	✗
Ferramenta	Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste	✓
	Disponibilidade de um banco de dados de teste (pacote de testes que simulam o ambiente de teste) para reutilização	✗
	Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos	✓
	Utilização de ferramentas para estimar esforço de teste e/ou cronograma	⚠
	Utilização de ferramentas de medida de cobertura nas fases de teste (unidade, integração, sistema)	✗
	Utilização de ferramentas geradoras de procedimentos ou casos de testes	✗

Pode ser observado que a maioria das práticas de teste de software atendidas parcialmente por Maraká está relacionada à organização dos testes dentro da empresa (Tabela 6.1), e correspondem a decisões da gerência. Sendo assim, elas não fazem parte do escopo de Maraká, porém podem ser aplicadas caso a gerência adote esta estratégia dentro da empresa.

Outras práticas de teste de software não foram atendidas por não fazerem parte do escopo de Maraká, que visa apoiar originalmente às atividades de planejamento e controle dos testes, como pode ser observado entre as práticas da categoria *Medição e Análise* e algumas práticas da categoria *Ferramenta*.

As demais práticas de teste de software analisadas são apoiadas por Maraká.

Com isso, atingiu-se o objetivo definido para este trabalho. O capítulo seguinte, "*Considerações Finais*", apresenta as conclusões deste trabalho, ressaltando os resultados obtidos, as suas limitações e próximos passos a serem realizados dando continuidade ao trabalho em questão.

Capítulo 7

Considerações Finais

Neste capítulo são descritas as contribuições obtidas a partir deste trabalho, as suas limitações e possíveis trabalhos futuros a serem realizados dando continuidade ao trabalho em questão.

7.1 – Visão Geral

Neste trabalho foi apresentada uma proposta de apoio ao planejamento e controle de testes de software, a partir da construção de uma infra-estrutura computacional que possibilita a aplicação de uma abordagem que apóia o planejamento e controle de testes composta por dois elementos básicos: *documentação* e *sistematização*.

Esta proposta foi elaborada visando a atender algumas das necessidades de organizações de software, a partir de informações obtidas em dois cenários: acadêmico, caracterizado por meio de revisões na literatura da área de teste de software; e prático, caracterizado por meio dos resultados obtidos em uma pesquisa de opinião que avaliou o estado atual das atividades de teste de software em um ambiente real de desenvolvimento, e ao final, identificou carências e limitações das organizações de software em relação à aplicação das práticas exclusivas que apóiam o planejamento e controle dos testes, conforme descrito no Capítulo 3. Com isso, duas expectativas principais foram criadas para que a infra-estrutura proposta, denominada Maraká, possa efetivamente trazer benefícios para o planejamento e controle de testes de software:

A primeira dessas expectativas é que a sua implantação possibilite a sistematização dos testes de software em organizações, a partir da utilização do processo de testes de software definido neste trabalho que servirá como guia para a realização das tarefas necessárias em uma atividade de teste.

A segunda expectativa é fornecer um apoio adequado para o controle e a documentação dos testes, como forma de evidenciar as atividades realizadas ao longo do processo de testes, permitindo aos envolvidos nessas atividades acompanhar os resultados obtidos, e podendo, ainda, reaproveitar essas informações em novos projetos.

7.2 – Contribuições

Nesta seção são descritas as principais contribuições identificadas neste trabalho:

Construção da infra-estrutura computacional para apoiar o planejamento e controle de teste software. A descrição do desenvolvimento e as funcionalidades da infra-estrutura Maraká estão descritas no Capítulo 5. Esta infra-estrutura pode ser aplicada em organizações de software para apoiar o planejamento e controle dos testes em seus projetos. Esta infra-estrutura:

- possui um conjunto de requisitos funcionais de apoio baseado nos resultados de um estudo que avaliou as necessidades reais de organizações de software em relação às atividades de teste de software.
- pode ser utilizada para planejar e controlar todas as atividades de teste de software, independente do tipo dos testes (unidade, integração, sistema, etc.), em uma organização, seguindo o processo de testes definido e gerando todos os artefatos desse processo;
- permite o acesso às informações das atividades de teste de acordo com os diferentes papéis associados ao processo de testes de software;
- pode ser utilizada para gerenciar as informações da equipe de teste em uma organização, alocando os seus membros em atividades de teste controladas pela infra-estrutura;
- Possibilita o controle dos testes por meio da visualização de gráficos que indicam o andamento e os resultados dos testes, cronogramas e relatórios de acompanhamento dos testes;
- pode ser utilizada por equipes geograficamente distribuídas, por estar disponível por meio da Internet, e possibilita o uso simultâneo de múltiplos usuários;
- está disponível nos idiomas português e inglês, mas possibilita a adição de novos idiomas por meio de um mecanismo de localização;

Pesquisa de Opinião para caracterização do estado da prática na área de teste de software. O Capítulo 3 descreve o planejamento, execução e os resultados obtidos em uma pesquisa de opinião para caracterização das práticas de teste que são utilizadas em um cenário de desenvolvimento de software brasileiro. Apesar dos seus resultados não

poderem ser generalizados para outros cenários, eles representam um ponto inicial, porém limitado, na caracterização das atividades de teste de software na indústria, e revelam diversas limitações e necessidades reais de organizações de software em relação às atividades de teste e servem como fundamento para este trabalho e para novas pesquisas na área de teste de software. O plano e a instrumentação do estudo foram definidos e possibilita a execução da pesquisa de opinião em outros cenários de desenvolvimento de software a fim de caracterizá-los e com isso obter um maior conhecimento sobre o cenário atual de teste de software em ambientes reais de desenvolvimento.

Definição de um processo de teste software. O Capítulo 4 descreve detalhadamente um processo de testes de software, como mecanismo para sistematização dos testes em uma organização. São definidos papéis, atividades (e suas dependências), critérios e artefatos a serem produzidos no processo. A partir da descrição deste processo, organizações podem implantá-lo em projetos reais, adaptando-o de acordo com suas características próprias (como nível de maturidade ou categoria de software desenvolvido) ou necessidades.

Revisão da literatura de teste de software. O Capítulo 2 descreve os principais conceitos relacionados ao planejamento e controle de teste de software. Além disso, a partir de uma revisão sistemática executada, foram identificadas e caracterizadas, a partir de critérios pré-estabelecidos, abordagens que apóiam o planejamento e controle de teste de software. Isso possibilita que outros pesquisadores possam re-executar o protocolo definido para identificar na literatura técnica abordagens que apóiam o planejamento e controle de testes de software,

7.3 – Limitações do Trabalho

Conforme observado na pesquisa de opinião descrita no Capítulo 3, o estado atual no cenário onde o estudo ocorreu indica que essas organizações de software possuem dificuldades em gerenciar suas atividades de testes. Além disso, a aplicação de Maraká em ambientes reais de desenvolvimento requer uma organização inicial das atividades de desenvolvimento de software, pois os artefatos produzidos ao longo do desenvolvimento são insumos necessários para o planejamento, execução e controle dos testes.

Sendo assim, hipoteticamente, a implantação de Maraká causa menos impacto em organizações que não sejam completamente imaturas em relação a práticas de Engenharia de Software e que estejam familiarizadas com a utilização de processos, pois o passo fundamental para implantação de Maraká é a utilização do processo de testes inserido na infra-estrutura. As demais organizações terão inicialmente dificuldades em utilizá-la, mas podem aplicá-la como uma estratégia para sistematização das suas atividades de teste.

Além dos fatores organizacionais relacionados à implantação de Maraká em ambientes reais de desenvolvimento, algumas características previstas para a infra-estrutura não puderam ser inicialmente atendidas, representando limitações do trabalho desenvolvido. São elas:

- Desenvolvimento de um mecanismo que possibilidade a integração de dados com ferramentas externas de apoio às atividades de geração de casos e procedimentos de teste, medição da cobertura dos testes e execução dos testes. Atualmente, todos os casos e procedimentos de teste devem ser especificados exclusivamente a partir de Maraká. O mesmo ocorre com os incidentes ocorridos durante os testes, que só podem ser registrados atualmente por Maraká.
- Desenvolvimento de um mecanismo que possibilidade a utilização de diferentes processos de testes por meio de Maraká, de forma que o responsável na organização possa escolher o processo a ser seguido em cada atividade de teste. Atualmente Maraká só possui um processo de testes disponível, que é o processo descrito no Capítulo 4 deste trabalho.
- Desenvolvimento de um mecanismo para reaproveitamento dos dados gerados em atividades de teste anteriores em novos testes, ou seja, reutilizar, por exemplo, os casos e procedimentos de teste de software em outros projetos.

7.4 – Trabalhos Futuros

Alguns trabalhos podem ser sugeridos como atividades futuras com o intuito de dar continuidade a este trabalho:

Relacionados à pesquisa de opinião para avaliação do estado da prática de teste de software:

- Aplicação da pesquisa de opinião em novos cenários de desenvolvimento de software, com o objetivo de avaliar os resultados obtidos nesses diferentes contextos e analisar o comportamento das práticas de teste de software em diferentes cenários reais de desenvolvimento de software.

Relacionados à infra-estrutura Maraká:

- Construção de um mecanismo que possibilite a integração de dados com ferramentas externas a Maraká, apoiando as atividades de identificação e construção de casos e procedimentos de teste e execução dos testes (automatização da execução dos casos e procedimentos de teste e registro dos incidentes de teste, que pode ser feito, por exemplo, a partir da ferramenta *BugZilla* [www.bugzilla.org]).
- Construção de um mecanismo que possibilite a utilização de diferentes processos de testes por Maraká, permitindo a escolha do processo a ser seguido em cada atividade de teste.
- Construção de um mecanismo para reutilização dos dados gerados em atividades de teste em novas atividades criadas por Maraká.
- Criação de um mecanismo que apóie as atividades de Medição e Análise dos testes, possibilitando uma melhoria constante das atividades de teste.
- Criação de um mecanismo para Gerência de Conhecimento em Maraká, permitindo que lições aprendidas sejam adicionadas e consultadas a qualquer momento.
- Realização de estudos experimentais com o objetivo de avaliar a viabilidade de utilização da infra-estrutura proposta em ambientes reais de desenvolvimento com relação ao apoio fornecido ao planejamento e controle de teste de software. Atualmente, Maraká está sendo aplicada em um projeto de desenvolvimento de software coordenado pelo Grupo de Engenharia de Software Experimental¹⁷ no contexto da COPPE-UFRJ.

¹⁷ Endereço da página do Grupo de Engenharia de Software Experimental: www.cos.ufrj.br/~ese.

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.
- ANDERSSON, C., RUNESON, P., "Verification and Validation in Industry - A Qualitative Survey on the State of Practice", In: International Symposium on Empirical Software Engineering (ISESE'02), p.37, October 03-04, 2002.
- BARCELLOS, M. P., "Planejamento de Custos em Ambientes de Desenvolvimento de Software Orientados à Organização", Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, julho, 2003.
- BASILI, V.R., CALDIERA, G., ROMBACH, H. D., "Goal Question Metric Paradigm". In J. J. Marciniak (ed.), Encyclopedia of Software Engineering, John Wiley & Sons, 1994.
- BEIZER, B., "Software testing techniques", 2nd ed., Van Nostrand Reinhold Co., New York, NY, 1990.
- BERTOLINO, A., "The (Im)maturity Level of Software Testing", SIGSOFT Softw. Eng. Notes 29, 5, Sep, 2004.
- BOOCH, G., RUMBAUGH, J., JACOBSON, I., "The Unified Modeling Language User Guide", 1 ed., Addison-Wesley, 1998.
- BRYAN, W. L., SIEGEL, S. C., "Software Product Assurance Techniques for Reducing Software Risk", Elsevier Science Publishing Co., New York, 1988.
- BURNSTEIN, I., SUWANASSART, T., Carlson, C. R., "Developing a Testing Maturity Model for Software Test Process Evaluation and Improvement", IEEE International Test Conference '96, Washington, DC, pp 581- 589, Oct, 1996.

CMMI PRODUCT DEVELOPMENT TEAM (2000): "CMMI-SE/SW: Capability Maturity Model – Integrated for Systems Engineering/Software Engineering", version 1.0 staged representation. Technical Report 2000-TR-012, Software Engineering Institute, Carnegie Mellon University, USA.

COSTA, H. R., "Uma Abordagem Econômica Baseada em Riscos para Avaliação de uma Carteira de Projetos de Software", Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, abril 2005.

CRAIG, R.D., JASKIEL, S. P., "Systematic Software Testing", Artech House Publishers, Boston, 2002.

CRESPO, A. N., SILVA, O. J., BORGES, C. A., SALVIANO, C. F., ARGOLLO, M., JINO, M., "Uma metodologia para teste de Software no Contexto da Melhoria de Processo", In: III Simpósio Brasileiro de Qualidade de Software (SBQS 2004), Brasília, 2004.

DENG, Y., HE, Z., "BATOOM: A Practical Approach to Testing Object-Oriented Software", Technology of Object-Oriented Languages and Systems (TOOLS), p. 328-337, September, 1998.

(a) DIAS NETO, A. C., TRAVASSOS, G. H., "Towards a Computerized Infrastructure for Software Testing Planning and Control". In: LATW 2005 - Latin American Test Workshop, 2005, Salvador. Proceedings of the LATW 2005, 2005.

(b) DIAS NETO, A.C., TRAVASSOS, G.H., "Uma Infra-estrutura para Apoiar o Planejamento e Controle de Teste de Software". In: III Workshop de Teses e Dissertações em Qualidade de Software (IV SBQS), 2005, Porto Alegre, RS.

FALBO, R., "Integração de Conhecimento em um Ambiente de Desenvolvimento de Software", Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil. , dez 1998.

FARIAS, L.D., "Planejamento de Riscos em Ambientes de Desenvolvimento de Software Orientados à Organização", Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Ago 2002.

FRANKL, P. G., WEISS, S. N., HU, C., "All-Uses vs Mutation Testing: An Experimental Comparison of Effectiveness", Journal of Systems and Software. Volume 38, Número 3, Pág. 235-253, Setembro 1997.

GELPERIN, D., HETZEL, B., "The Growth of Software Testing", Communications of the ACM, Vol. 31(6), p. 687-695, 1988.

HETZEL, B., "The Systematic Test and Evaluation Process Summary Guide", Software Quality Engineering, 1985.

HOWDEN, W.E., "Functional program testing and analysis". Nova York, NY, McGrall-Hill, 1987.

IEEE Standard 1008-1987, IEEE Standard for Software Unit Testing, IEEE Press.

IEEE Standard 1012-1986, IEEE Standard for Software Verification and Validation, IEEE Press.

IEEE Standard 1059-1993, IEEE Guide for Software Verification and Validation Plans, IEEE Press.

IEEE Standard 610-1990: IEEE Standard Glossary of Software Engineering Terminology, IEEE Press.

IEEE Standard 829-1983: Standard for Software Test Documentation, IEEE Press.

IEEE Standard 829-1998: Standard for Software Test Documentation, IEEE Press.

ISO/IEC 12207. "Information Technology – software life cycle process". International Standard Organization, 1995.

ISO/IEC 15504-5. "Information Technology – Process Assessment – Part 5: Guidance on use for Process Improvement and Process Capability Determination", Geneve: ISO, 2003.

ISO/IEC 15504-5. "Information Technology – Process Assessment – Part 5: An Exemplar Process Assessment Model", Geneva: ISO, 2004.

JAMIELSON, S., "Likert scales: how to (ab)use them", *Med Educ* 2004, 38:1217-1218.

JURISTO, N., MORENO, A. M., VEGAS, S., "Reviewing 25 years of testing technique experiments". *Empirical Software Engineering: An International Journal*, 9(1), p. 7-44, March 2004.

KANER, C., BACH, J., "Rapid test planning", *Software Testing, Analysis & Review Conference (STAR) West*, San Jose, CA, October, 2001.

KIM, J. M., PORTER, A., ROTHERMEL, G., "An Empirical Study of Regression Test Application Frequency". In *Proceedings of the 22nd International Conference of Software Engineering*, Maio 2000, Pág. 126-135. Ireland. IEEE Computer Society Press.

KITCHENHAM, B., "Procedures for Performing Systematic Reviews", *Joint Technical Report Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia Ltd*, 2004, Australia.

KOOMEN, T., POL M., "Test Process Improvement: A practical step-by-step guide to structured testing", Addison-Wesley, 1999.

LAITENBERGER, O., VEGAS, S., CIOLKOWSKI, M., "The State of the Practice of Review and Inspection Technologies in Germany", *Report Visek 011/E*, 2002.

MAFRA, S. N., TRAVASSOS, G.H., "Técnicas de Leitura de Software: Uma Revisão Sistemática", *In: XIX Simpósio Brasileiro de Engenharia de Software*, Uberlândia, 2005.

MATS, L., "The top five software-testing problems and how to avoid them", *EDN Europe*, Feb2001, Vol. 46 Issue 2, p37, 3p; (AN 4121152).

- MCGREGOR, J. D., SYKES, D. A., "A practical guide to testing object-oriented software", Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2001.
- MENDONÇA, C. C., "Uma Infra-Estrutura para Apoio ao Planejamento e Execução de Pesquisas de Opinião na Web", Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2005.
- MPS.BR – Melhoria de Processo de Software Brasileiro, Guia Geral (v. 1.0). 2005.
- MYERS, G., "The Art of Software Testing", New York: John Wiley, 1979.
- PERRY, W., "Effective Methods for Software Testing", Wiley, 1995.
- PFLEEGER, S. L., "Engenharia de Software – Teoria e Prática", Prentice Hall, 2ª Edição, São Paulo, 2004.
- Project Management Institute, "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)", 3th Edition, 2004.
- PRESSMAN, R. S., "Software Engineering: A Practitioner's Approach", McGraw-Hill, 6th ed, Nova York, NY, 2005.
- RAPPS, S., WEYUKER, E.J., "Data Flow analysis techniques for test data selection", In: International Conference on Software Engineering, p. 272-278, Tokio, Sep. 1982.
- ROCHA, A. R. C., MALDONADO, J. C., WEBER, K. C., "Qualidade de software – Teoria e prática", Prentice Hall, São Paulo, 2001.
- SAMSON, D., "Knowledge-Based Test Planning: Framework for a Knowledge-Based System to Prepare a System Test Plan from System Requirements", J. Systems Software, 1993, pp.115-124
- SCHNAIDER, LR.C., "Planejamento da Alocação de Recursos Humanos em Ambientes de Desenvolvimento de Software Orientados à Organização", Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, maio 2003.

SEPIN (Secretaria de Política de Informática), MCT (Ministério da Ciência e Tecnologia), Qualidade E Produtividade no Setor de Software - Resultados da Pesquisa, 2005. <http://www.mct.gov.br/sepim/Dsi/Quali2005/Public2005.htm>.

SOFTWARE ENGINEERING INSTITUTE (SEI), "CMM for Software", Version 1.1, SEI-93-TR-24 and -25, 1993.

SEI/CMU – SOFTWARE ENGINEERING INSTITUTE/CARNEGIE MELLON UNIVERSITY, "ISO 15504 - Frequently Asked Questions (FAQs)", <http://www.sei.cmu.edu/cmimi/faq/15504-faq.html>, acessado em 20/04/2006.

VAN OSSEL, G., 1998, "Measuring customer satisfaction". In: Looy, B., Van Dierdonck, R., Gemmel, P. (eds), *Services Management: an integrated approach*, chapter 8, Great Britain, Pitman Publishing.

VILLELA, K., "Definição e Construção de Ambientes de Desenvolvimento de Software Orientados à Organização", Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, maio 2004.

XEXEÉO, J.A.M. (2001) "Sistemas de Informação como Instrumento de Programas de Qualidade", Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, dezembro.

WOHLIN, C., RUNESON, P., HÖST, M., OHLSSON, M., REGNELL, B., WESSLÉN, A., 2000, "Experimentation in Software Engineering – An Introduction", Kluwer Academic Publishers.

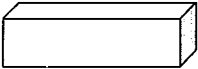



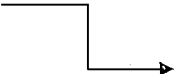


Anexo A

Notação usada na Modelagem do Processo de Testes de Software

Este anexo apresenta a notação utilizada na modelagem do processo de testes de software definido neste trabalho.

A linguagem proposta para modelagem de processos descrita em (VILLELA, 2004) é composta de elementos gráficos que podem ser do tipo área, objeto ou ligação, onde uma ligação estabelece uma relação entre dois objetos e uma área agrupa objetos, definindo um contexto para os mesmos. Objetos ainda permitem adornos, utilizados para representar explicitamente características dos objetos. A seguir, cada elemento da linguagem que foi utilizado neste trabalho é brevemente apresentado. Esses elementos foram descritos na seção 4.4.1 deste trabalho.

Tabela A.1 – Entidade e Forma de Representação

Entidade	Forma de Representação
Processo	
Ator	
Atividade Atômica	
Atividade Composta	
Dependência entre Atividades	
Documento	
Fluxo de Entrada/Saída	

Apêndice A

Questionários Aplicados na Pesquisa de Opinião para Caracterização de Práticas de Revisão e Teste de Software

Este apêndice apresenta os questionários utilizados durante a execução da pesquisa de opinião descrita neste trabalho.

É importante ressaltar que o estudo, originalmente, foi conduzido com o objetivo de caracterizar práticas de revisão e teste de software, e não somente de teste de software, mas no contexto deste trabalho somente os resultados dos itens relacionados a teste de software foram apresentados e analisados. Sendo assim, foram desenvolvidos e aplicados três questionários na pesquisa realizada.

O primeiro, *Formulário de Caracterização da Organização ou Unidade Organizacional*, visa à caracterização das organizações nas quais os participantes do estudo atuam. Este questionário possui itens relacionados à localização, categoria de softwares desenvolvidos, domínio das aplicações, tamanho da organização, além da caracterização das atividades de Garantia da Qualidade.

O segundo, *Formulário de Caracterização do Participante*, visa à caracterização dos participantes do estudo. Este questionário possui itens relacionados à posição do indivíduo na organização, sua formação e grau de experiência em Revisão e Teste de Software.

O terceiro, *Formulário com Práticas de Revisão e Teste de Software*, é o questionário com os itens a serem avaliados no estudo. Os itens estão divididos em duas partes: a primeira são itens relacionados às atividades de revisão de software, e a segunda parte possui itens relacionados a teste de software.

Os questionários aplicados na pesquisa de opinião, juntamente com o Plano do Estudo podem ser obtidos para consulta e futuras aplicações por meio do pacote do estudo (conjunto de documentos) produzido neste trabalho. O pacote do estudo pode ser obtido contatando os pesquisadores responsáveis por este trabalho.

- Arilo Cláudio Dias Neto: acdnd@cos.ufrj.br.
- Guilherme Horta Travassos: ght@cos.ufrj.br.

Apêndice B

Dados Coletados na Pesquisa de Opinião para Caracterização de Práticas de Revisão e Teste de Software

Este apêndice apresenta os dados coletados durante a execução da pesquisa de opinião descrita neste trabalho.

Ao longo da execução da pesquisa de opinião foi coletado um conjunto amplo de dados, referentes às respostas do Formulário de Caracterização da Organização, Formulário de Caracterização do Participantes e *Formulário com Práticas de Revisão e Teste de Software*.

As tabela com os dados obtidos contém as seguintes informações:

- **ORGANIZAÇÃO:** Identificador da organização que foi visitada durante o estudo. Número criado para manter todas as organizações anônimas.
- **ID:** Identificador do participante do estudo em relação à organização em que atua. Número criado para manter todos os participantes do estudo anônimos.
- **IDADE (ANOS):** Idade da organização desde a sua fundação.
- **IDADE (SW):** Tempo que a organização atua no desenvolvimento de software.
- **QP:** Quantidade de projetos de software que o participante atuou até o momento.
- **TA (ANOS):** Tempo de atuação em desenvolvimento de software.
- **Formação:** Nível de formação acadêmica do participante o estudo. Os valores possíveis são:
 0. Nível Técnico;
 1. Nível Superior;
 2. Especialização;
 3. Mestrado;
 4. Doutorado;

- **Exp Ver:** Nível de experiência nas atividades de Verificação. Os valores possíveis são:
 0. Nenhum;
 1. Baixo;
 2. Médio;
 3. Alto;
 4. Excelente;
- **Exp Val:** Nível de experiência nas atividades de Validação. Os valores possíveis são:
 0. Nenhum;
 1. Baixo;
 2. Médio;
 3. Alto;
 4. Excelente;
- **PT:** Peso Total obtido para um participante do estudo.
- **ORG:** Identificador da organização no estudo. Cada organização recebeu um identificador único.
- **Atividades (A1 a A16):** conjunto de itens que compuseram o estudo e que estão relacionados às atividades de teste de software. Esses itens podem ser obtidos no Apêndice A deste trabalho.
- **Processo (P1 a P11):** conjunto de itens que compuseram o estudo e que estão relacionados ao processo de teste de software. Esses itens podem ser obtidos no Apêndice A deste trabalho.
- **Ferramentas (F1 a F6):** conjunto de itens que compuseram o estudo e que estão relacionados às ferramentas de apoio aos testes de software. Esses itens podem ser obtidos no Apêndice A deste trabalho.
- **Os valores de cada célula representam as possíveis escalas para avaliação da aplicabilidade, conforme descrito a seguir:**

Em relação à aplicação das práticas de revisão ou teste de software em uma organização, as opções foram:

 0. Não Aplicável – não faz parte do contexto da organização.
 1. Não Utilizado pela organização.
 2. Uso Não Freqüente: Usado poucas vezes.
 3. Uso Habitual: Usado em grande parte dos projetos de software da organização.

4. Uso Padrão: Usado em todos os projetos de software da organização.

Em relação ao grau de importância das práticas de revisão e teste de software em uma organização, as opções serão:

0. Não Importante: Não necessário.
1. Valor Baixo: Pouco importante.
2. Valor Limitado: Poderia ser adequado utilizá-lo.
3. Valor Significante: Prática recomendada.
4. Valor Essencial: Deve ser uma prática padrão para todos.

As tabelas com os dados obtidos no estudo podem ser obtidos contatando os pesquisadores responsáveis por este trabalho.

- Arilo Cláudio Dias Neto: acdn@cos.ufrj.br.
- Guilherme Horta Travassos: ght@cos.ufrj.br.

Apêndice C

Roteiros dos Documentos de Teste descritos no IEEE *Standard 829*

Este apêndice apresenta os roteiros dos documentos descritos no IEEE Standard 829 e que foi utilizado neste trabalho.

Os documentos do IEEE Standard 829 (1998) citados neste Apêndice foram utilizados no Processo de Testes de Software definido no Capítulo 4 e são produzidos pela infra-estrutura Maraká, descrita no Capítulo 5.

Utilizando a infra-estrutura Maraká, a equipe de testes pode editar cada roteiro (*template*) de documento que compõe o processo de testes de software. Os roteiros utilizados representam soluções iniciais para documentação dos testes propostas neste trabalho, mas podem ser adaptados às necessidades de cada organização.

Os roteiros definidos neste trabalho são referentes aos seguintes documentos:

- Plano de Testes;
- Especificação de Projeto de Teste;
- Especificação de Caos de Teste;
- Especificação de Procedimento de Teste;
- Histórico de Teste;
- Relatório de Incidente de Teste;
- Relatório de Resumo de Teste;

Os roteiros de cada um dos documentos listados anteriormente podem ser obtidos contatando os pesquisadores responsáveis por este trabalho.

- Arilo Cláudio Dias Neto: acdn@cos.ufrj.br.
- Guilherme Horta Travassos: ght@cos.ufrj.br.