

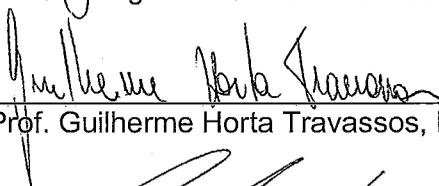
UMA ABORDAGEM PARA AVALIAÇÃO DE PROPOSTAS DE  
MELHORIA EM PROCESSOS DE SOFTWARE

Reinaldo Cabral Silva Filho

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. Ana Regina Cavalcanti da Rocha, D.Sc.

  
\_\_\_\_\_  
Prof. Guilherme Horta Travassos, D.Sc.

  
\_\_\_\_\_  
Prof. Geraldo Bonorino Xexeo, D.Sc.

  
\_\_\_\_\_  
Prof. Márcio de Oliveira Barros, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MAIO DE 2006

SILVA FILHO, REINALDO CABRAL

Uma Abordagem para Avaliação de  
Propostas de Melhoria em Processos de  
Software [Rio de Janeiro] 2006

XI, 167 p. 29,7 cm (COPPE/UFRJ, M.Sc.,  
Engenharia de Sistemas e Computação, 2006)

Dissertação - Universidade Federal do Rio de  
Janeiro, COPPE

1. Melhoria de Processos de Software

2. Experimentação

I. COPPE/UFRJ II. Título (série)

À Izalina e Reinaldo, meus amados pais.

## **Agradecimentos**

---

À minha mãe, pelo amor que me permite superar todos os óbices e intensifica meus momentos de conquista e felicidade.

Ao meu pai, que me inspira e fomenta a minha criatividade.

Às minhas irmãs, Isabela e Eveline, pelo afago permanente em meu coração, estímulo constante e vitórias compartilhadas.

À Mylene Lisbôa, pela compreensão, paciência, contribuição e pelo seu amor.

Aos meus tios Benedito, Hélio, Irenita, Jaime, Marlene, Marlise e Vanda, pelo apoio em momentos difíceis e presença nos momentos felizes.

À Luciana Lima, pela fantástica demonstração de amizade em momentos críticos.

À Máira Paula e Simone Lara, amigas de todas as horas.

À família Lisbôa, pelo incentivo.

À Tayana Conte, pela amizade e companheirismo.

Aos novos amigos Adriano, Ahilton, Ana Cândida, Andrea, Arilo, Fábio, Gleison, Jeann, Marco, Marcos, Mariano, Paula, Paulo, Rafael, Rodrigo, Sávio, Sômulo e Wladimir, por participarem e agregarem valor a esta experiência sensacional (!!!).

Aos meus orientadores, Ana Regina Rocha e Guilherme Travassos, pela oportunidade, apreço, respeito e confiança depositada.

À Equipe TABA, pelo conhecimento, cooperação e pelo tratamento que me fez sentir em casa.

Ao Grupo de Engenharia de Software Experimental, pelas discussões empolgantes e profundas reflexões que colaboraram com o desenvolvimento da minha visão enquanto eterno aprendiz.

À cooperativa MAXMA, em especial, aos amigos Adriano, Alex Henrique, Alex Müller, Carlos Henrique, Guilherme, Gustavo, Josué, Mylene (é ela mesmo!), Niedson e Renata, pela participação, contribuição e amizade fortalecida ao longo dos anos.

Aos amigos Evandro de Barros Costa e Arturo Hernández Domínguez, pela crença em meu potencial.

À empresa BL Informática, pela participação e à Analia Irigoyen pela atenção e colaboração.

À Jucele Vasconcellos, pela pronta cooperação.

À equipe da Secretaria do Programa e à Cláudia Prata, pela atenção dispensada às questões triviais e não triviais do cotidiano.

À Taísa Guidini, pela presteza no trato das questões operacionais.

À Universidade Federal de Alagoas, pela oportunidade de evoluir profissionalmente.

Ao CNPQ, 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 ABORDAGEM PARA AVALIAÇÃO DE PROPOSTAS DE MELHORIA EM PROCESSOS DE SOFTWARE

Reinaldo Cabral Silva Filho

Maio/2006

Orientadores: Ana Regina Cavalcanti da Rocha

Guilherme Horta Travassos

Programa: Engenharia de Sistemas e Computação

As organizações que fazem uso de processos como base para o desenvolvimento de seus produtos, frequentemente se deparam com a necessidade de realizar alterações para adequá-los às especificidades do negócio e os objetivos gerais da organização. A partir de suas experiências, as organizações podem identificar e adaptar aquelas práticas mais ajustadas à sua identidade e incorporá-las em seus processos. Medições de produto, comparações entre resultados de projetos e dados históricos, bem como análise de tendências podem auxiliar na avaliação dos potenciais efeitos das mudanças nos processos de software.

Contudo, vários fatores podem dificultar a visualização dos reais efeitos das mudanças realizadas nos processos de software em prol da melhoria, dentre eles: a qualidade dos dados mensurados, a forma como a mudança foi introduzida e observada, a preparação do ambiente para a mudança e outros. Desconsiderar estes aspectos significa estar suscetível a interpretações equivocadas, o que pode orientar a organização na direção errada.

Neste contexto, esta Dissertação propõe uma abordagem para a realização de projetos-piloto no ambiente industrial de forma sistemática, planejada e controlada para que os efeitos das melhorias em processos de software possam ser observados e mensurados adequadamente antes da institucionalização das mudanças.

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

AN APPROACH TO EVALUATION OF  
SOFTWARE PROCESS IMPROVEMENT PROPOSALS

Reinaldo Cabral Silva Filho

May/2006

Advisors: Ana Regina Cavalcanti da Rocha

Guilherme Horta Travassos

Department: System and Computing Engineering

Organizations that use software processes as a base to product development often need change and adapt their process to accommodate new business goals. From past experiences, organizations can identify and tailor the best practices to their context. Measurements, benchmarking and tendency analysis are important tools to evaluate the potential effects of software process changes.

Many factors can difficult the real effects verification of the changes in benefit of software improvement, such as: incorrect measure collection, the way changes were introduced and observed, environment preparation, among others. Disregarding these aspects can lead to misinterpretations, which can cause the organization to move in the wrong direction.

In this context, this thesis presents an approach for performing pilot projects within an industrial environment in a systematic, planned and controlled manner in order to observe and measure the real effects of software process changes before its institutionalization.

# Conteúdo

---

Capítulo 1. Introdução .....	1
1.1. Motivação .....	1
1.2. Objetivos.....	2
1.3. Breve Histórico .....	3
1.4. Organização da Dissertação .....	4
Capítulo 2. Avaliação da Efetividade das Propostas de Melhoria de Processo de Software .....	6
2.1. Introdução.....	6
2.2. Processos de Software .....	6
2.3. Melhoria de Processos de Software .....	7
2.4. Modelos em Prol da Melhoria de Processos de Software .....	10
2.4.1. Quality Improvement Paradigm (QIP).....	10
2.4.2. IDEAL.....	12
2.4.3. ISO/IEC 15504 .....	13
2.4.4. Capability Maturity Model Integration (CMMI) .....	17
2.4.5. Modelo de Referência para Melhoria de Processos de Software (MR-MPS) .....	19
2.5. Instrumentos para Avaliar a Efetividade das Melhorias em Processos de Software.....	21
2.5.1. Estudos experimentais.....	21
2.5.2. Simulação .....	24
2.5.3. Benchmarking .....	26
2.5.4. Pilotos .....	26
2.5.5. Combinação de abordagens.....	28
2.6. Modelos em Prol da Qualidade de Software e o Uso de Projetos-Piloto .....	29
2.7. Considerações Finais.....	30
Capítulo 3. Ambientes de Engenharia de Software Centrados em Processo e a Estação TABA .....	32
3.1. Introdução.....	32
3.2. Ambientes de Engenharia de Software Centrados em Processos (AESCPs) .....	33
3.3. A Estação TABA.....	34
3.3.1. Breve histórico .....	34
3.3.2. Visão geral da abordagem para definição de processos .....	36
3.3.3. Estratégia de avaliação e melhoria em níveis da Estação TABA.....	37
3.4. AESCP e a Avaliação da Efetividade das Propostas de Melhoria.....	38
3.5. Considerações Finais.....	39
Capítulo 4. Instrumentos que Apóiam a Realização de Pilotos – Um Estudo Baseado em Revisão Sistemática .....	40
4.1. Introdução.....	40
4.2. Estratégia Utilizada .....	40
4.3. Desenvolvimento do Protocolo de Pesquisa.....	41

4.4. Desenvolvimento dos Instrumentos de Apoio.....	43
4.5. Execução do Estudo .....	44
4.6. Análise dos Dados Coletados .....	50
4.7. Considerações Finais.....	51
4.7.1. Fatores que podem ter influenciado a pesquisa.....	51
4.7.2. Questões em aberto e algumas hipóteses .....	51
4.7.3. Lições aprendidas .....	52
4.7.4. Conclusões .....	52
Capítulo 5. Uma Abordagem Baseada em Experimentação para Avaliação da Efetividade de Propostas de Melhoria de Processos de Software .....	54
5.1. Introdução.....	54
5.2. A Abordagem Baseada em Experimentação.....	54
5.3. O Processo.....	55
5.3.1. Elicitação das atividades do processo.....	56
5.3.2. Adaptação das atividades elicitadas às necessidades da indústria .....	57
5.3.3. Modelo do processo.....	62
5.4. A Ferramenta Pilot .....	68
5.5. O Guia .....	69
5.6. Considerações Finais.....	70
Capítulo 6. Pilot, Uma Ferramenta para Apoiar a Realização de Pilotos .....	72
6.1. Introdução.....	72
6.2. O Processo de Desenvolvimento.....	72
6.2.1. Planejamento inicial .....	72
6.2.2. Requisitos .....	74
6.2.3. Projeto.....	77
6.2.4. Construção.....	81
6.2.5. Implantação.....	82
6.3. Plano para Realização de um Piloto da Abordagem .....	82
6.3.1. Identificação do piloto .....	83
6.3.2. Melhoria a ser avaliada.....	83
6.3.3. Plano de medição .....	83
6.3.4. Expectativas para a avaliação .....	84
6.3.5. Critérios de seleção .....	85
6.3.6. Fatores que podem influenciar a avaliação .....	86
6.4. Execução do Piloto da Abordagem.....	86
6.5. Resultados Parciais do Piloto .....	89
6.5.1. Análise quantitativa parcial .....	89
6.5.2. Fatores de influência identificados até o momento .....	89
6.5.3. Análise qualitativa parcial .....	90
6.6. Considerações Finais.....	91

Capítulo 7. Aplicação da Abordagem na Indústria – Um Piloto na BL Informática .....	93
7.1. Introdução.....	93
7.2. A BL Informática.....	94
7.3. Planejamento do Piloto na BL Informática .....	94
7.3.1. Identificação do piloto .....	94
7.3.2. Descrição da melhoria a ser avaliada.....	96
7.3.3. Elaboração do plano de medição .....	96
7.3.4. Registro das expectativas para o piloto .....	98
7.3.5. Critérios para seleção de projetos .....	98
7.3.6. Identificação dos fatores de influência.....	99
7.3.7. Emissão do plano do piloto e do documento de diretrizes .....	100
7.4. Execução do Piloto na BL Informática .....	102
7.4.1. Associação do projeto ao piloto.....	102
7.4.2. Monitoração do piloto.....	102
7.4.3. Execução do projeto-piloto .....	104
7.5. Considerações Finais.....	104
Capítulo 8. Considerações Finais.....	105
8.1. Conclusão.....	105
8.2. Perspectivas Futuras.....	107
REFERÊNCIAS BIBLIOGRÁFICAS .....	108
APÊNDICE A - Protocolo do estudo baseado em revisão sistemática .....	117
APÊNDICE B - Fontes das publicações identificadas na seleção preliminar .....	123
APÊNDICE C - Publicações obtidas a partir de mais de uma biblioteca digital.....	125
APÊNDICE D - Publicações selecionadas para filtragem.....	126
APÊNDICE E - Publicações com restrições de acesso nas bibliotecas digitais .....	138
APÊNDICE F - Publicações selecionadas no 1º filtro .....	140
APÊNDICE G - Modelo de processo para realização de projetos-piloto .....	142
ANEXO I - Notação para Modelagem de Processos .....	148
ANEXO II – Plano para realização do piloto elaborado pela BL Informática .....	151
ANEXO III – Diretrizes para realização do piloto na BL Informática .....	159
ANEXO IV – Registro de monitoração do piloto na BL Informática .....	167

# Índice de figuras

---

Figura 2.1 - Ciclo PDCA (Plan, Do, Check, Act) (DEMING, 1982).....	8
Figura 2.2 - Quality Improvement Paradigm (MCGARRY, 2002).....	11
Figura 2.3 - O modelo IDEAL versão 1.1 (GREMBA e MYERS, 1997) .....	13
Figura 2.4 - O processo de avaliação e os principais contextos de utilização (ISO/IEC, 2003).....	14
Figura 2.5 - Processo de melhoria na ISO/IEC 15504 (ISO/IEC, 2003). .....	16
Figura 2.6 - Origem e evolução do CMMI (KULPA; JOHNSON, 2003). .....	18
Figura 2.7 - Estrutura do MR-MPS (SOFTEX, 2005).....	20
Figura 2.8 - Experimento: um fator e três tratamentos .....	24
Figura 3.1 - Estrutura da Estação TABA definida por Travassos (1994) .....	35
Figura 3.2 - Abordagem para definição de processos na Estação TABA (VILLELA et al., 2004).....	36
Figura 3.3 - Estratégia para avaliação e melhoria de processos em níveis (CAMPOS, 2005). .....	38
Figura 4.1 - Modelo entidade-relacionamento do banco de dados de apoio à revisão sistemática. ....	44
Figura 4.2 - Formulário para apoiar a extração dos dados. ....	44
Figura 5.1 - Visão macro do processo .....	63
Figura 5.2 - Modelo do Processo para Realização de Pilotos .....	67
Figura 5.3 - Interação do processo com outras atividades e processos da organização.....	68
Figura 5.4 - Diagrama de casos de uso com as funcionalidades da Pilot.....	69
Figura 6.1 - Estrutura Organizacional do Projeto.....	73
Figura 6.2 - Casos de uso representando a interação do grupo de processo com a abordagem .....	74
Figura 6.3 - Casos de uso representando a interação do gerente e equipe do projeto com a abordagem	75
Figura 6.4 - Protótipo da tela para descrever as ações de melhoria.....	76
Figura 6.5 - Versão corrente da tela para descrição das ações de melhoria.....	76
Figura 6.6 - Primeira versão do diagrama de classes.....	77
Figura 6.7 - Diagrama de estados do piloto.....	77
Figura 6.8 - Diagrama de classes.....	78
Figura 6.9 - Diagrama de estados do projeto-piloto.....	79
Figura 6.10 - Visão arquitetural da integração da ferramenta Pilot com a infra-estrutura provida pela Estação TABA .....	80
Figura 6.11 - Interação da ferramenta Pilot com aplicações que compõem a Estação TABA .....	81
Figura 7.1 - Tela Identificar Piloto.....	95
Figura 7.2 - Tela Descrever Melhorias para Avaliação.....	96
Figura 7.3 - Tela Elaborar Plano de Medição.....	97
Figura 7.4 - Tela Registrar Expectativas da Avaliação.....	98
Figura 7.5 - Tela Estabelecer Critérios para Seleção do Projeto.....	99
Figura 7.6 - Tela Identificar Fatores de Influência.....	100
Figura 7.7 - Plano para Realização do Piloto.....	101
Figura 7.8 - Diretrizes para o gerente do projeto.....	101
Figura 7.9 - Projeto BL 00177/2006 instanciado na Estação TABA e associado ao piloto.....	103
Figura 7.10 - Tela Identificar Piloto após a associação do projeto BL 00177/2006 ao piloto.....	103

## Índice de Tabelas

---

Tabela 2.1 - Avaliação de níveis de capacidade da ISO/IEC 15504 (ISO/IEC, 2003). .....	15
Tabela 2.2 - Níveis de maturidade e áreas de processo (CHRISIS, 2003). .....	19
Tabela 2.3 - Níveis de maturidade, processos e capacidade do MR-MPS (SOFTEX, 2005). .....	20
Tabela 4.1 - Práticas para realização de pilotos sugeridas por modelos de qualidade. ....	42
Tabela 4.2 - Bibliotecas digitais selecionadas. ....	45
Tabela 4.3 - Expressões de busca e opções aplicadas nas bibliotecas digitais. ....	45
Tabela 4.4 - Total de documentos por biblioteca – seleção e catalogação preliminar.....	46
Tabela 4.5 - Total de publicações selecionadas na segunda etapa por fonte. ....	47
Tabela 4.6 - Publicações selecionadas para extração de dados.....	48
Tabela 4.7 - Registro preliminar das publicações com dados para coleta.....	48
Tabela 4.8 - Publicações selecionadas e características dos respectivos instrumentos de apoio à realização de pilotos .....	49
Tabela 4.9 - Totalização de publicações, fontes e bibliotecas por etapa.....	50
Tabela 5.1 - Conjunto de atividades elicitadas. ....	56
Tabela 6.1 - Outras partes envolvidas no projeto. ....	73
Tabela 6.2 - Cronograma inicial.....	73
Tabela 6.3 - Disponibilidade de recursos para o projeto.....	74
Tabela 6.4 - Critérios do piloto e o perfil da organização.....	87
Tabela 6.5 - Registros de monitoração do piloto. ....	89

# Capítulo 1. Introdução

---

## 1.1. Motivação

As organizações que fazem uso de processos como base para o desenvolvimento de seus produtos frequentemente se deparam com a necessidade de realizar alterações para adequá-los às especificidades do negócio e os objetivos gerais da organização. A partir de avaliações e experiências bem conduzidas, as organizações podem identificar e adaptar aquelas práticas mais ajustadas a sua identidade e incorporá-las em seus processos. Medições de produto e processo são usadas para avaliar a efetividade das melhorias. Comparações entre resultados de projetos-piloto e dados históricos, bem como tendências gerais podem testar e demonstrar os efeitos da mudança do processo. (WALRAD e MOSS, 1993).

Neste contexto, Kelley e Morath (2001) ilustram três situações: (i) Uma mudança é realizada, porém nenhum dado foi coletado antes ou depois da mudança para avaliar se os efeitos foram positivos ou negativos; (ii) A mudança é realizada, mas os dados que caracterizam a situação anterior à mudança não são coletados; e (iii) Há dados coletados antes e depois da mudança que permitem verificar se os efeitos foram positivos ou não.

Mesmo no terceiro caso, algumas questões precisam ser consideradas durante a análise dos efeitos da mudança: Os dados coletados estão corretos? Existem outros fatores que podem dificultar a interpretação dos resultados? Os efeitos observados podem ser generalizados? A mudança foi conduzida de forma correta a ponto de apresentar os reais efeitos? As pessoas foram devidamente preparadas para a mudança? A mudança poderia ser melhor? Desconsiderar estes aspectos significa estar suscetível a interpretações equivocadas, o que pode orientar a organização na direção errada.

Por outro lado, considerar todas estas questões é uma tarefa não trivial. Na academia, pesquisadores fazem uso do paradigma experimental para colher evidências para constituir o corpo de conhecimento em Engenharia de Software. Neste contexto, dois obstáculos oferecem resistência à transferência tecnológica dos achados para a indústria: a ausência de controle nas experiências em ambiente industrial e a ausência de realismo na realização de experimentos controlados. Além disso, é necessário convencer a indústria sobre a validade e aplicabilidade dos resultados (SJØBERG et al, 2002).

Modelos que tratam da melhoria de processos de software, tais como IDEAL (McFeeley, 1996), QIP (Basili et al., 1994), CMMI (CHRISISS et al., 2003) e MR-MPS (SOFTEX, 2005), sugerem o uso de pilotos quando há riscos e incertezas associadas à efetividade das ações. Porém, estes modelos não mencionam procedimentos que permitam avaliar de maneira consistente os efeitos obtidos a partir das mudanças dos processos. Dibã et al. (2005) afirmam que a avaliação da efetividade é um aspecto crítico, porém um ponto fraco da melhoria de processos de software na indústria.

Bill Curtis (2000) afirma que todos os modelos de melhoria, de forma geral, sugerem cinco passos: (i) escolha um problema importante, (ii) projete uma solução, (iii) pilote a solução; (iv) revise a solução e (v) dissemine a solução por toda a organização. Entretanto, o único problema é que a equipe de ação, responsável pela introdução das melhorias, fica estagnada durante muito tempo no passo três, enquanto o restante da organização “pega fogo”. Em outras palavras, é preciso mais objetividade, reduzir o consumo de esforço para avaliar as propostas de melhoria e aumentar a acurácia destas avaliações para, o quanto antes, disseminar as melhorias na organização.

O estudo apresentado em Silva Filho et al. (2005) forneceu indícios de que a realização de projetos-piloto pode prover clareza e objetividade no tratamento das questões supracitadas.

Este trabalho propõe a realização de projetos-piloto de forma sistemática, planejada e controlada para que os efeitos das melhorias em processos de software possam ser observados e mensurados adequadamente antes da institucionalização das mudanças.

## **1.2. Objetivos**

O principal objetivo desta dissertação é apresentar uma abordagem para apoiar a avaliação da efetividade das propostas de melhoria de processo de software. A abordagem, baseada em experimentação, permite a realização de projetos-piloto de forma sistemática, planejada e controlada para que os efeitos das mudanças dos processos de software possam ser observados e mensurados antes de sua institucionalização.

São elementos da abordagem: um processo, uma ferramenta de software e um guia. O processo contém todas as atividades necessárias para realizar a avaliação das propostas de melhoria. A ferramenta de software serve de instrumento de apoio à execução do processo e se integra à ambientes de engenharia de software centrados

em processo. O guia provê conhecimento necessário para que a abordagem seja utilizada adequadamente.

### **1.3. Breve Histórico**

Em meados de 2004, no curso de Engenharia de Software Experimental foram apresentados e discutidos diversos estudos publicados na área. A experimentação foi posta como um instrumento valioso a ser utilizado para compreender e avaliar tecnologias, métodos e técnicas nos mais diversos cenários. A dificuldade de realizar experimentos In Vivo, ou seja, no ambiente industrial e com uma população real, foi uma das questões abordadas. Essa dificuldade reside principalmente na ausência de controle sobre o ambiente, no tamanho reduzido da população que participa do estudo, nos riscos envolvidos e no custo.

Concomitantemente, o Grupo de Qualidade de Software da COPPE/UFRJ trabalhava na construção de uma estratégia de avaliação e melhoria que considerasse os múltiplos níveis relacionados à definição, especialização e instanciação de processos. Na ocasião houve um aprofundamento nas discussões a respeito da necessidade de melhoria contínua dos processos de software e da importância desta no aumento da capacidade e maturidade das organizações. A estratégia de avaliação e melhoria estava sendo definida para apoiar a melhoria de processos e compor o conjunto de instrumentos provido às organizações pela Estação TABA.

Então, a partir das discussões e trabalhos realizados naquele momento, foi proposto um estudo que permitisse avaliar a viabilidade de aplicar uma abordagem experimental em uma pequena ou média empresa de software com o intuito de avaliar a efetividade das mudanças nos processos de software e, conseqüentemente, minimizar os riscos associados a essas alterações<sup>1</sup>.

A partir desse estudo, realizado na indústria em 2004, foram identificadas as dificuldades de avaliar a efetividade das melhorias executadas em um processo. No estudo experimental para avaliar os efeitos obtidos com as melhorias realizadas no processo, ficou evidente a necessidade de elaborar procedimentos específicos baseados em experimentação para avaliação de melhoria de processos em um ambiente industrial. Também foi perceptível a carência de mecanismos que apoiassem o planejamento, a operação e a análise do estudo ora realizado.

Considerando as lacunas identificadas e o contexto industrial de software, que tem provocado uma demanda por mecanismos que apóiem a melhoria contínua de

---

<sup>1</sup> O relato da experiência consta em Silva Filho et al. (2005).

seus processos, foi delineada a proposta de uma abordagem que atendesse a esse cenário.

A proposta inicial deste trabalho foi apresentada no Workshop de Teses e Dissertações em Qualidade de Software de 2005, evento satélite do SBQS 2005. O relato da experiência na indústria foi apresentado na mesma ocasião, no evento principal. O produto desta dissertação foi submetido como trabalho técnico no SBQS 2006 e será apresentado no evento.

## **1.4. Organização da Dissertação**

A dissertação está organizada em oito capítulos, incluindo este capítulo introdutório.

O segundo capítulo, Avaliação da Efetividade das Propostas de Melhoria de Processos de Software, apresenta uma visão geral relacionada à melhoria de processos de software. Modelos, normas e algumas abordagens são discutidas sobre a perspectiva da avaliação da efetividade das mudanças realizadas nos processos de software.

No terceiro capítulo, um estudo, baseado em revisão sistemática, é apresentado com o intuito de identificar instrumentos que têm sido propostos para facilitar a realização de pilotos no âmbito industrial com o objetivo de avaliar a efetividade das mudanças dos processos de software.

O quarto capítulo descreve a abordagem que está sendo proposta, seus elementos constituintes e como eles interagem para facilitar a realização de pilotos na indústria.

O quinto capítulo expõe as características de ambientes de engenharia de software centrados em processos e discorre sobre o desenvolvimento e evolução da Estação TABA. Finalizando o capítulo, é realizada uma breve discussão de como estes ambientes centrados em processos podem contribuir para a avaliação da efetividade das propostas de melhoria de processos de software.

O sexto capítulo trata dos aspectos mais relevantes do desenvolvimento do elemento ferramenta da abordagem, a ferramenta Pilot. Dentre eles: o planejamento inicial, projeto, a construção e implantação. O capítulo também relata o planejamento e o início de um piloto da abordagem com a intenção de observar a utilidade e a potencial contribuição do instrumento à melhoria de processos de software.

O sétimo capítulo ilustra o uso da abordagem no ambiente industrial. Um piloto é planejado através da interação com a ferramenta Pilot e é iniciado para avaliar a

efetividade de uma nova versão de processo com relação à diminuição do custo com a qualidade no desenvolvimento de projetos de pequeno porte.

Concluindo a Dissertação, são apresentados os aspectos tratados pela proposta, apontadas as contribuições e descritas as questões a serem abordadas futuramente.

## **Capítulo 2. Avaliação da Efetividade das Propostas de Melhoria de Processo de Software**

---

### **2.1. Introdução**

De acordo com a percepção de qualidade de um produto, o processo de desenvolvimento de software pode ser modificado para que o resultado final ressalte determinada característica. Entretanto, para realizar estas alterações, é imperativo que antes o processo esteja definido. Só então, após ser observado e mensurado, é possível melhorá-lo (SHULL et al., 2001).

Modelos em prol da qualidade como o Quality Improvement Paradigm (QIP) (BASILI et al., 1994) e o CMMI (CHRISISS, 2003) apóiam a identificação de oportunidades de melhoria dos processos de software e sugerem que as ações de melhoria sejam avaliadas antes de disseminá-las por toda a organização.

O objetivo deste capítulo é apresentar instrumentos que podem ser utilizados para avaliar a efetividade dessas propostas de melhoria antes que estas sejam institucionalizadas. Uma breve discussão sobre o papel dos ambientes de engenharia de software, no que tange ao apoio à melhoria de processos, também é apresentada.

### **2.2. Processos de Software**

Na indústria de software, o desenvolvimento e a manutenção de produtos são atividades complexas, influenciadas por fatores técnicos, culturais e ambientais. Esta complexidade tem impacto direto no custo, prazo, esforço e na qualidade do produto. Sistematizar a execução das atividades, através de processos de software viabiliza a repetição, independente de quem as execute (PFLEEGER, 2004, p. 37), a mensuração e a identificação de oportunidades de melhoria.

Algumas definições de processo de software:

“Uma seqüência de passos executadas para um determinado propósito; por exemplo, o processo de desenvolvimento de software.” (IEEE, 1990)

“Um conjunto de atividades, métodos, práticas e transformações que pessoas utilizam para desenvolver e manter software e produtos associados (planos de projeto, código, casos de teste, manuais do usuário e outros)” (PAULK et al., 1993).

“Um processo de software pode ser definido como um conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos que é

necessário para conceber, desenvolver, implantar e manter um produto de software” (FUGGETTA, 2000)

A definição de um processo é feita através da construção de um modelo de processo. Um modelo de processo de software é uma representação prescritiva das atividades de desenvolvimento em termos da sua ordem de execução e gerenciamento dos recursos (OSTERWEIL, 1987); consiste na descrição de um processo de software que permite que o mesmo seja analisado, compreendido e executado. Chrissis *et al.* (2003) enfatizam que a descrição de um processo é uma expressão documentada de um conjunto de atividades executadas para alcançar um dado propósito, que provê uma definição operacional dos principais componentes de um processo.

Modelos e normas em prol da qualidade dos processos de software como CMMI (CHRISSIS, 2003), MR-MPS (SOFTEX, 2005), ISO/IEC 15504 (ISO/IEC, 2003) e ISO/IEC 12207 (ISO/IEC, 1995) servem de guia para apoiar a definição de processos. Eles sugerem atividades e, em alguns casos, produtos que devem ser gerados. Contudo, não descrevem os processos a serem seguidos. A definição destes processos é responsabilidade da organização e depende de muitos fatores, incluindo o domínio da aplicação, a estrutura e o tamanho da organização (CHRISSIS *et al.*, 2003).

### **2.3. Melhoria de Processos de Software**

O princípio básico da melhoria de processos está calcado no ciclo PDCA (Figura 2.1), proposto por Walter Shewart da década de 20 e difundido amplamente na indústria japonesa após a segunda guerra por um de seus seguidores, William Edwards Deming. O ciclo se inicia a partir da definição dos objetivos de melhoria e planejamento das atividades de melhoria (P - Plan). Em seguida, o plano é posto em prática (D - Do), os resultados obtidos com as ações de melhoria são verificados (C - Check) e, por fim, ações de refinamento e ajuste são realizadas no processo e o ciclo se reinicia.

O contexto do desenvolvimento de software apresenta diferenças quando comparado ao ambiente da manufatura de produtos. Contudo, na opinião de Fuggetta (2000), as iniciativas de melhoria em processos de software também devem considerar as contribuições de cientistas organizacionais para minimizar o risco de reinventar estratégias e ignorar questões importantes que podem influenciar de forma significativa a iniciativa de melhoria, pois “melhoria de processo de software também é melhoria de processos” (Ibidem, p. 32). Isto pode ser claramente observado nos

fundamentos de modelos como IDEAL (MCFEELEY, 1996), CMMI (CHRISISS, 2003), ISO/IEC 15504 (ISO/IEC, 2003) e MR-MPS (SOFTEX, 2005) que, inspirados no ciclo PDCA, sugerem a evolução gradativa e contínua dos processos de software.



Figura 2.1 - Ciclo PDCA (Plan, Do, Check, Act) (DEMING, 1982).

Outra característica observada em modelos de qualidade de processo de software é o estabelecimento de níveis de maturidade e capacidade. Para Paulk et al. (1993b) a maturidade da organização de software é determinante para o alcance dos seus objetivos. Em organizações imaturas, processos de software são geralmente improvisados e, mesmo que um processo de software tenha sido especificado, ele não é rigorosamente seguido. Cronogramas e orçamentos são rotineiramente excedidos por serem baseados em estimativas não realistas. Quando prazos são impostos, a funcionalidade do produto e a qualidade são freqüentemente comprometidas para atender ao cronograma. Já em organizações maduras, gerentes comunicam, de forma eficaz, o processo de software à equipe e aos novos membros. As atividades são executadas de acordo com o plano do processo e são consistentes com a forma que o trabalho é feito. Os processos definidos são atualizados sempre que necessário, com melhorias desenvolvidas através de testes pilotos e análises de custo-benefício. Há julgamento da qualidade do produto e análise dos problemas do processo e produto de forma quantitativa. Cronogramas e orçamentos são realistas e baseados no histórico de desempenho. Os resultados esperados para custo, cronograma, funcionalidades e qualidade são geralmente alcançados. A maturidade das organizações pode ser avaliada através de modelos de avaliação, a exemplo do SCAMPI (CMU/SEI, 2001) e da ISO/IEC 15504 (ISO/IEC, 2003), que auxiliam na

caracterização do estado da prática e na identificação dos pontos fortes e fracos para subsidiar ações de melhoria.

Existem diversas abordagens para prover melhoria aos processos de software que, de uma forma geral, podem ser agrupadas em abordagens *bottom-up* e *top-down* (MCGARRY; THOMAS, 1994). Em uma abordagem *top-down* a melhoria é considerada como uma mudança realizada nos processos para que se aproximem a um padrão genérico e amplamente aceito pelo grau de qualidade que pode prover à organização. Em uma abordagem *bottom-up* a melhoria é obtida a partir da experiência e do conhecimento adquirido em projetos anteriores e de acordo com os produtos, objetivos e características da organização. As mudanças são realizadas considerando um escopo local e não práticas aceitas universalmente. Nesta categoria encontramos o *Quality Improvement Paradigm* (QIP) (BASILI et al., 1994) e a Fábrica de Experiências (BASILI; CALDIERA, 1995).

A escolha da abordagem de melhoria mais adequada não é tarefa trivial. É possível, inclusive, combiná-las para constituir uma estratégia que considere as boas práticas dos modelos de qualidade e o contexto ao qual a organização está inserida. Conradi e Fuggetta (2002) afirmam que é difícil identificar fatores que possam servir de critério para selecionar uma abordagem de melhoria, dada a existência de estudos ilustrando casos de sucesso e benefícios ao utilizar determinadas abordagens e outros estudos que contradizem ou minimizam estes benefícios. A partir da experiência e de estudos realizados na indústria de software da Noruega, os autores fazem seis proposições para dirimir este problema:

i) A escolha do modelo deve apoiar melhorias com foco nas metas e na inovação de produtos. A disciplina e a estabilidade necessárias para obter um nível maior de maturidade não devem servir de óbices para o alcance de metas de curto prazo e à contínua inovação;

ii) Sempre que possível, deve-se optar por iniciativas *bottom-up*. Para isso, priorizam-se as necessidades mais urgentes e simples, estimula-se o envolvimento dos desenvolvedores, coleta-se o feedback a cada iniciativa e torna-o visível à alta gerência;

iii) O uso de ferramentas com o propósito de automatizar o processo deve ser cauteloso. Elas são úteis em processos não criativos, como gerência de configuração e inspeção. Contudo, para atividades criativas é preciso garantir que a ferramenta seja adaptável e adequada ao contexto;

iv) O uso de modelos que sejam capazes de quantificar o retorno do investimento feito na iniciativa de melhoria deve ser estimulado;

v) A promoção da mudança cultural deve ser gradual, com ações que minimizem o impacto e estimulem o aprendizado e a participação no processo;

vi) É importante realizar experimentos para entender como as pessoas trabalham e é interessante estabelecer um sistema de gratificação para relatos de problemas e sugestões de melhorias.

Rifkin (2001) advoga que o sucesso da iniciativa de melhoria depende da estratégia de negócio adotada pela organização e o alinhamento da abordagem de melhoria a esta estratégia.

## **2.4. Modelos em Prol da Melhoria de Processos de Software**

Nesta seção serão apresentados alguns modelos que servem como guia para apoiar iniciativas em prol da melhoria de processos de software.

### **2.4.1. Quality Improvement Paradigm (QIP)**

O Quality Improvement Paradigm (BASILI, 1989; BASILI e CALDIERA, 1995; BASILI et al., 1994) é resultado da aplicação do método científico ao problema da melhoria da qualidade do software. Briand et al. (1995) consideram o QIP como uma instância do método científico para a engenharia de software. O QIP, inspirado no ciclo PDCA, é baseado em uma abordagem cíclica e contínua para melhoria de processos que consiste em seis passos:

(1) **Caracterize.** Utilize modelos, dados, intuição e quaisquer mecanismos úteis ao entendimento do ambiente. Estabeleça linhas base com os processos de negócio existentes na organização e caracterize-os quanto à criticalidade;

(2) **Estabeleça objetivos.** Com base na caracterização inicial e nos aspectos de relevância estratégica para a organização, estabeleça objetivos quantificáveis para projetos bem sucedidos, desempenho e melhoria da organização;

(3) **Escolha o processo.** Com base na caracterização do ambiente e dos objetivos que foram determinados, escolha os processos apropriados para submetê-los a melhorias, ferramentas e métodos de apoio, certificando-se de que eles são consistentes com os objetivos que foram estabelecidos;

(4) **Execute.** Execute os processos nos projetos, analise os dados obtidos em cada projeto e forneça um retorno a respeito dos dados que estão sendo coletados em prol do alcance dos objetivos;

(5) Análise. Ao final de cada projeto específico, analise os dados da informação reunida para avaliar as práticas atuais, determinar os problemas, registrar os achados e realizar recomendações para futuras melhorias dos projetos;

(6) Empacote e armazene a experiência. Consolide a experiência reunida na forma de novo ou atualizado e refinado modelo ou em outras formas de conhecimento estruturado, reunido a partir dos projetos realizados. Armazene a experiência em uma base de experiência para torná-la disponível para futuros projetos.

O QIP implementa dois ciclos (Figura 2.2). O primeiro ciclo, ciclo de retorno do projeto (ciclo controlado), provê retorno na forma de indicadores quantitativos do projeto. Estes dados são úteis para prevenir e solucionar problemas. O segundo ciclo, ciclo de retorno corporativo (ciclo de capitalização), provê um retorno à organização com dois propósitos: (i) prover informação analítica sobre a execução dos projetos finalizados através da comparação dos dados do projeto com a faixa nominal da organização e analisar concordâncias e discrepâncias; e (ii) acumular experiência reutilizável na forma de artefatos de software que são aplicáveis a outros projetos.

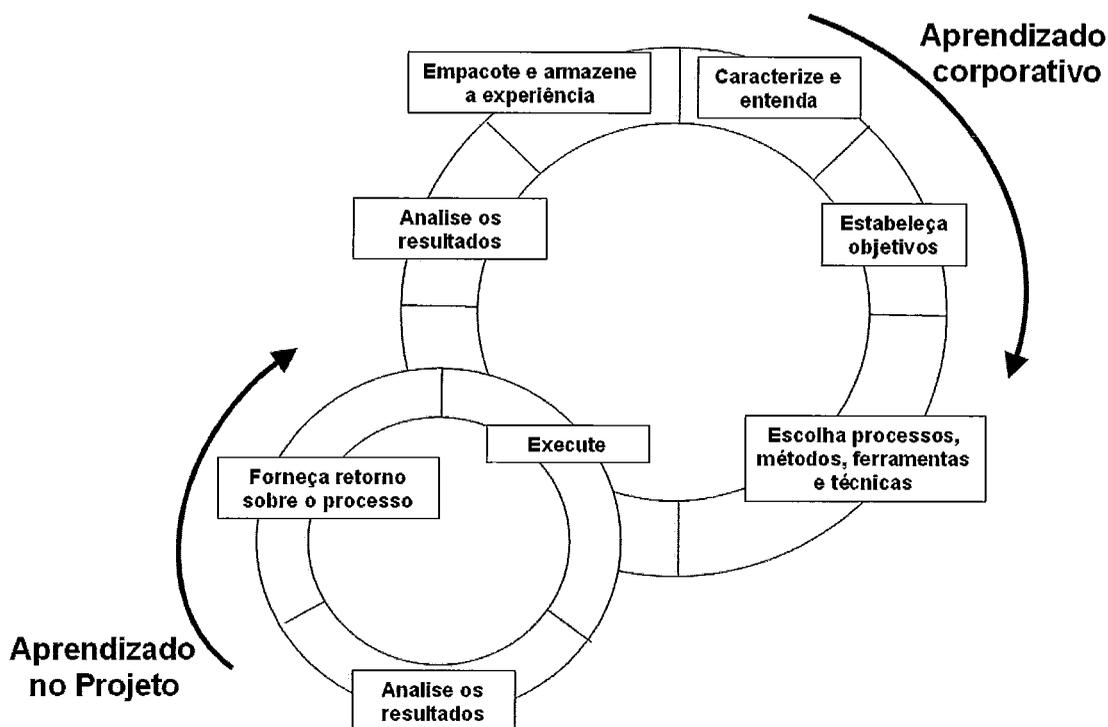


Figura 2.2 - Quality Improvement Paradigm (MCGARRY, 2002).

No QIP a medição de software é considerada um componente indispensável para capturar experiências e recuperar conhecimento nas atividades de desenvolvimento. O paradigma sugere o método GQM como mecanismo de medição para definir e avaliar os objetivos de melhoria (MCGARRY, 2002).

## 2.4.2. IDEAL<sup>2</sup>

Mcfeeley (1996) descreve o IDEAL como um modelo de programa organizacional de melhoria de processos de software que serve como guia para iniciar, planejar e implementar ações de melhoria. Segundo Gremba e Myers (1997), o propósito inicial do modelo era descrever, em termos gerais, os estágios que uma organização envolvida com melhoria de processos de software deveria passar. Em 1997, uma nova versão do modelo (Figura 2.3) foi desenvolvida para ser aplicável não apenas na melhoria de processos de software, mas a qualquer esforço de melhoria.

O modelo IDEAL é composto por cinco fases: Iniciar, Diagnosticar, Estabelecer, Executar e Aprender, descritas a seguir:

**Iniciar.** Nesta fase, a motivação e o esforço necessário para alcance das metas e objetivos de negócio são identificados, bem como a relação desse esforço com outros trabalhos da organização. O apoio da alta gerência é garantido e os recursos são alocados em ordem de magnitude. Por fim, uma infra-estrutura para gerenciar detalhes da implementação é estabelecida;

**Diagnosticar.** Nesta segunda fase são realizadas duas caracterizações: o estado atual da organização e o estado pretendido. A partir destas duas visões são realizadas recomendações para condução das atividades subsequentes. Estas recomendações são fundamentais nas tomadas de decisão dos gerentes e patrocinadores;

**Estabelecer.** Com base nas recomendações e nas restrições do ambiente organizacional, as prioridades são estabelecidas. Tendo em vista o escopo do trabalho, os fatores técnicos e não-técnicos, as habilidades e o conhecimento requerido, uma abordagem é desenvolvida para combinar a execução do trabalho com a disponibilidade de recursos. Uma vez definida a abordagem, um plano detalhado é elaborado, contendo cronograma, atividades, marcos, pontos de decisão, recursos, responsabilidades, planos de medição, mecanismos de rastreabilidade, riscos e estratégias de mitigação e outros elementos requeridos pela organização;

**Executar.** Na fase Executar, o trabalho que foi idealizado e planejado nas fases anteriores é executado. Uma melhor solução para atender as necessidades da organização é criada. Esta solução combina elementos chave, tais como processos, ferramentas, conhecimento, habilidades, informação e apoio externo. Em seguida, a

---

<sup>2</sup> IDEAL é um acrônimo das fases compõem a versão inicial do modelo proposto por Mcfeeley (1996): *Initiating* (Iniciar), *Diagnosing* (Diagnosticar), *Establishing* (Estabelecer), *Acting* (Executar) e *Leveraging* (Institucionalizar). Na versão 1.1 (GREMBA; MYERS, 1997), a última fase foi renomeada para *Learning* (Aprender) mantendo o acrônimo.

solução deve ser testada, através da execução de pilotos ou outros mecanismos que possibilitem identificar problemas e refinar a solução. Este refinamento pode consumir várias iterações até que a solução alcance um nível satisfatório. Após o refinamento, a solução é implantada na organização através de uma estratégia traçada de acordo com a natureza da melhoria e as circunstâncias organizacionais;

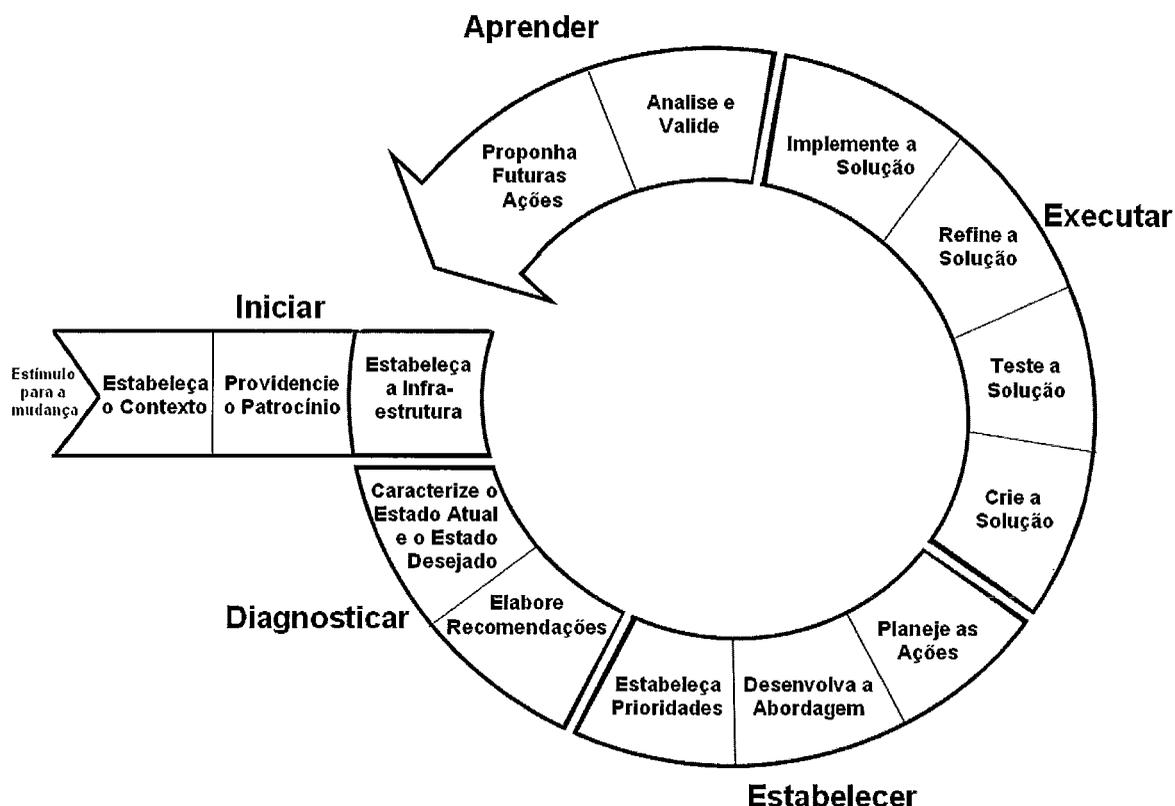


Figura 2.3 – O modelo IDEAL versão 1.1 (GREMBA e MYERS, 1997)

**Aprender.** Esta fase completa o ciclo de melhoria. A experiência é revisada para determinar se o esforço realizado atendeu os objetivos estabelecidos e como a organização pode implementar a mudança de forma mais eficiente e/ou eficaz no futuro.

### 2.4.3. ISO/IEC 15504

A International Organization for Standardization (ISO), em 1991, ao observar a proliferação de modelos em prol da qualidade e a deficiência da ISO 9001 no tratamento da qualidade relacionada à produção de software, decidiu investigar a necessidade da criação de um padrão internacional. A partir dos resultados desta investigação, a ISO criou em 1993 o projeto SPICE<sup>3</sup>. O objetivo era produzir um relatório técnico que fosse, ao mesmo tempo, mais geral e abrangente que os modelos

<sup>3</sup> Acrônimo para *Software Process Improvement and Capability dEtermination*.

existentes e mais específico que a ISO 9001 (CÔRTEZ; CHIOSSI, 2001). A última versão do relatório técnico foi aprovada em 1998. Este relatório técnico evoluiu e, em 2003, foi formalizado como norma internacional ISO/IEC 15504 (ISO/IEC, 2003).

A ISO/IEC 15504, centrada na avaliação de processos de software, é aplicável a dois contextos: para determinação da capacidade dos processos de software e para sistematizar a realização de programas de melhoria de processos de software (Figura 2.5). Desta forma, a norma visa apoiar, de forma distinta e complementar, a determinação dos pontos fortes e fracos da organização, a identificação dos riscos relacionados com os objetivos de negócio da organização e a melhoria contínua da eficiência e eficácia da organização.

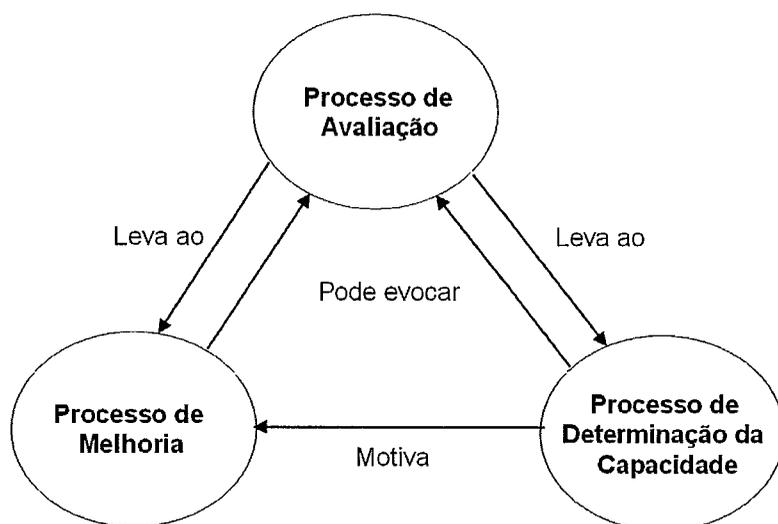


Figura 2.4 - O processo de avaliação e os principais contextos de utilização (ISO/IEC, 2003)

A norma, como arcabouço para determinação da capacidade dos processos, provê uma abordagem estruturada com os seguintes propósitos: entender o estado dos processos para melhoria do processo, determinar a adequação dos processos para um requisito particular ou classe de requisitos e determinar a adequação dos processos de uma organização para fins de contratação. A norma fornece resultados que caracterizam o estado atual dos processos da organização através da aferição dos níveis de capacidade (Tabela 2.1). Para a determinação da capacidade, a ISO/IEC 15504 provê um conjunto de requisitos para a avaliação e para o modelo a ser avaliado (parte 2, normativa), um guia para interpretar os requisitos para realizar a avaliação (parte 3, informativa) e um exemplo de modelo de avaliação de processo, tomando como base a ISO/IEC 12207 Amd 1 e Amd 2. (ISO/IEC, 1995, 2002, 2004) (parte 5, informativa).

**Tabela 2.1 - Avaliação de níveis de capacidade da ISO/IEC 15504 (ISO/IEC, 2003).**

Escala	Atributos do Processo	Avaliação
Nível 1	Execução do Processo	Largamente ou totalmente
Nível 2	Execução do Processo Gerência da Execução Gerência dos Produtos de Trabalho	Totalmente Largamente ou totalmente Largamente ou totalmente
Nível 3	Execução do Processo Gerência da Execução Gerência dos Produtos de Trabalho Definição do Processo Implantação do Processo	Totalmente Totalmente Totalmente Largamente ou totalmente Largamente ou totalmente
Nível 4	Execução do Processo Gerência da Execução Gerência dos Produtos de Trabalho Definição do Processo Implantação do Processo Medição do Processo Controle do Processo	Totalmente Totalmente Totalmente Totalmente Totalmente Largamente ou totalmente Largamente ou totalmente
Nível 5	Execução do Processo Gerência da Execução Gerência dos Produtos de Trabalho Definição do Processo Implantação do Processo Medição do Processo Controle do Processo Inovação do Processo Otimização do Processo	Totalmente Totalmente Totalmente Totalmente Totalmente Totalmente Totalmente Largamente ou totalmente Largamente ou totalmente

Quanto ao processo de melhoria, a ISO/IEC 15504 sugere a execução de oito passos (Figura 2.5):

**Passo 1: Examine os objetivos de negócio da organização.** Em geral os objetivos estão centrados na satisfação dos clientes, no aumento de competitividade ou no aumento de valor agregado dos produtos e serviços fornecidos. Tendo em vista estes objetivos de negócio, deve-se selecionar o modelo de referência mais adequado. O programa de melhoria de processo, que contém as estratégias, políticas, objetivos e as atividades relacionadas com o alcance de objetivos específicos de melhoria, é claramente estabelecido e entendido;

**Passo 2: Inicie o ciclo de melhoria de processo.** O programa de melhoria de processo deve ser implementado como um projeto, isto é, com a definição de cronogramas, responsabilidades, marcos e demais atributos. É desenvolvido um plano contendo: a descrição de como a mudança progressiva do processo será implementada, as atividades de monitoração e controle e as necessidades de treinamento para implementação das mudanças;

**Passo 3: Avalie a capacidade atual.** A capacidade do processo é avaliada conforme descrito no processo de determinação da capacidade da norma (parte 4);

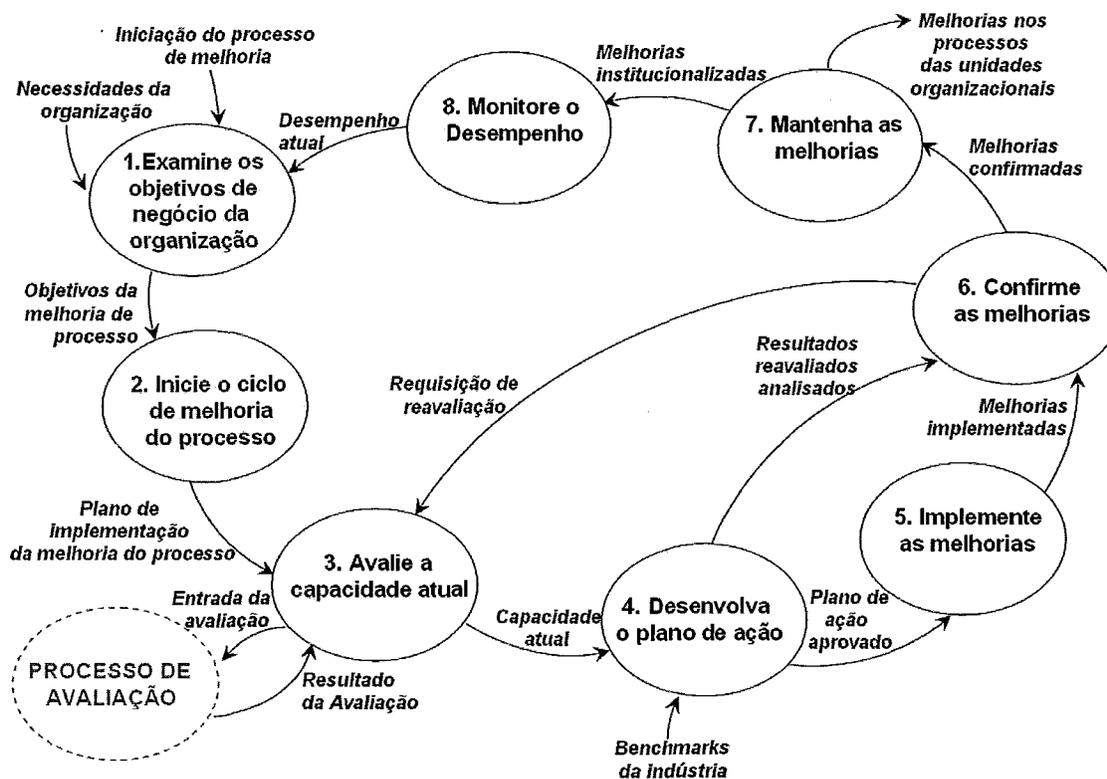


Figura 2.5 - Processo de melhoria na ISO/IEC 15504 (ISO/IEC, 2003).

**Passo 4: Desenvolva o plano de ação.** Com base nos resultados da avaliação, uma lista de áreas para melhoria é identificada e analisada, objetivos específicos de melhoria são estabelecidos como alvo e um plano contendo as ações de melhoria é traçado;

**Passo 5: Implemente as melhorias.** De acordo com a complexidade das melhorias, projetos de melhoria podem ser conduzidos, sempre que necessários, com o intuito de implementar uma ou mais ações. As quatro principais atividades de implementação do projeto de melhoria são: (i) selecionar a estratégia de melhoria, que pode consistir na realização de pilotos em uma unidade organizacional ou implementar a melhoria integralmente na organização. A escolha da estratégia depende do risco, do contexto organizacional e da disponibilidade de recursos; (ii) Detalhar o plano de implementação, que consiste na elaboração de um plano completo do projeto; (iii) Implementar ações de melhoria, executar o plano propriamente dito levando em conta fatores culturais, a manutenção do comprometimento, a comunicação com a equipe de trabalho e as necessidades de treinamento; e (iv) Monitorar a implementação, conforme estabelecido no plano do projeto de melhoria;

**Passo 6: Confirme as melhorias.** Após a implementação dos projetos de melhoria, a organização deve confirmar se os objetivos de melhoria foram alcançados, se os processos e práticas adequadas foram adotadas e se a cultura organizacional foi alterada de forma apropriada. Deve-se considerar a possibilidade de requerer uma

nova avaliação de capacidade para confirmar se a capacidade desejada foi estabelecida;

Passo 7: **Mantenha as melhorias.** Uma vez que a melhoria foi confirmada é necessário manter o novo nível de capacidade. Isto requer o monitoramento da institucionalização do processo melhorado através de medições apropriadas. Se a melhoria foi constatada em apenas uma unidade organizacional através de um piloto, a implantação deve atingir todas as áreas ou projetos da organização onde a melhoria for aplicável. A implantação deve ser apropriadamente planejada e documentada como parte do Plano do Programa de Melhoria de Processo;

Passo 8: **Monitore o desempenho.** A execução do processo deve ser monitorada através de um conjunto de métricas alinhadas com os objetivos de negócio. Novas melhorias devem tomar como base as experiências adquiridas a partir de experiências anteriores.

#### **2.4.4. Capability Maturity Model Integration (CMMI)**

O CMMI surgiu a partir da necessidade de integrar os modelos de maturidade propostos pelo SEI (engenharia de sistemas, engenharia de software, aquisição de software, gerência de força de trabalho e desenvolvimento integrado de processo e produto, dentre os mais relevantes). As diferenças entre esses modelos, principalmente no que tange a arquitetura, conteúdo e abordagem, estavam limitando a habilidade das organizações em empreender amplas iniciativas de melhoria de forma bem sucedida (CHRISISS, 2003). Assim, o CMMI foi elaborado a partir de três modelos: o Capability Maturity Model for Software (SW-CMM) v2.0 draft C, Systems Engineering Capability Model<sup>4</sup> (SECM) e o Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98. Estes modelos foram selecionados devido à ampla adoção da comunidade de engenharia de software e de sistemas e suas diferentes abordagens para melhorar os processos de software em uma organização. A Figura 2.6 ilustra a origem e a evolução do modelo.

O modelo reúne 25 áreas de processo, agrupadas em quatro níveis de maturidade e classificadas em quatro categorias (Tabela 2.2).

---

<sup>4</sup> Também conhecido como Electronic Industries Alliance Interim Standard (EIA/IS) 731 (EIA, 1998).

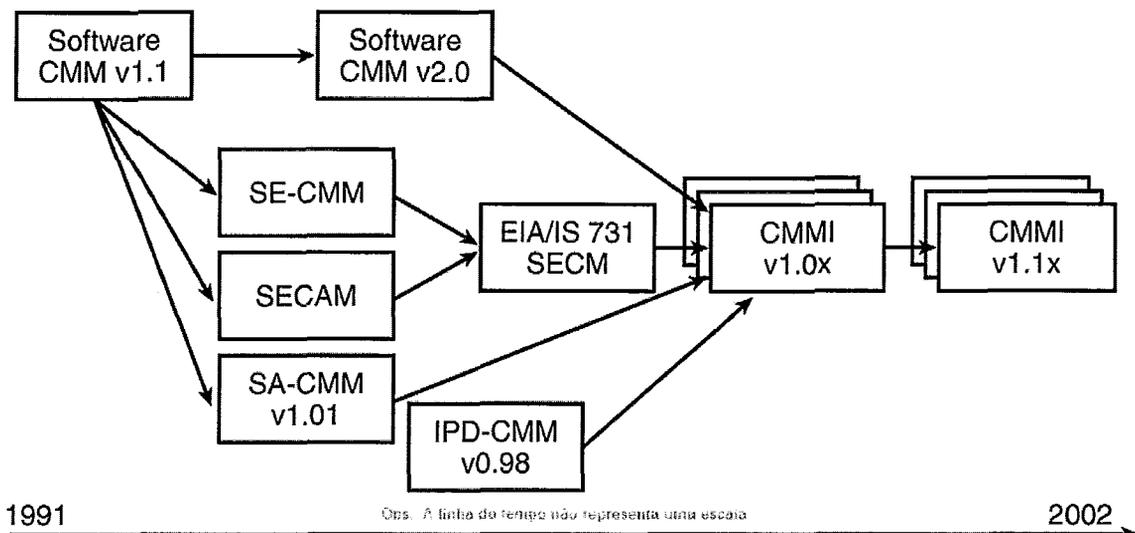


Figura 2.6 - Origem e evolução do CMMI (KULPA; JOHNSON, 2003).

Duas representações para utilização do modelo são oferecidas: por estágios ou contínuo. No contínuo, o foco da iniciativa é orientado às áreas de processo de interesse. As organizações selecionam as áreas de processo mais alinhadas com seus objetivos de negócio e empreendem em iniciativas de melhoria dos processos em prol do atendimento aos objetivos e práticas dos processos relacionados com estas áreas. Na representação por estágios, o foco da melhoria é a baseada no alcance de níveis de maturidade organizacional. Para cada nível de maturidade, é requerido que a organização implemente o conjunto de objetivos e práticas relacionadas às áreas de processo do nível pretendido. A obtenção de determinado nível de maturidade está condicionada à obtenção do nível de capacidade equivalente em todas as áreas de processo do nível pretendido. Por exemplo, para uma organização obter o nível de maturidade três, ela terá que atingir o nível de capacidade 3 para todas as áreas de processo relacionadas com os níveis 2 e 3.

Ambas as representações requerem o atendimento cumulativo às práticas e objetivos do nível de capacidade ou maturidade almejado, isto é, além do atendimento integral dos objetivos e práticas no nível pretendido, aqueles dos níveis inferiores ao pretendido também devem ser atendidos.

Vale ressaltar que as áreas de processo propostas pelo CMMI não possuem uma relação um para um com os processos definidos para a organização. A reunião dos objetivos e práticas por área de processo é apenas uma tentativa de agrupá-los por área de conhecimento. As organizações são responsáveis por definir seus processos conforme sua cultura e objetivos de negócio. Desta forma, um processo organizacional pode atender objetivos de uma ou mais áreas de processo do CMMI.

**Tabela 2.2 - Níveis de maturidade e áreas de processo (CHRISIS, 2003).**

Níveis	Áreas de processo	Categoria
Nível 2 (Gerenciado)	Gerência de Requisitos	Engenharia
	Planejamento do Projeto	Gerência de Projeto
	Monitoração e Controle do Projeto	Gerência de Projeto
	Gerência de Acordos com Fornecedores	Gerência de Projeto
	Medição e Análise	Suporte
	Garantia da Qualidade do Processo e do Produto	Suporte
Nível 3 (Definido)	Gerência de Configuração	Suporte
	Desenvolvimento de Requisitos	Engenharia
	Solução Técnica	Engenharia
	Integração do Produto	Engenharia
	Verificação	Engenharia
	Validação	Engenharia
	Foco no Processo Organizacional	Gerência de Processo
	Definição do Processo Organizacional	Gerência de Processo
	Treinamento Organizacional	Gerência de Processo
	Gerência de Projeto Integrada	Gerência de Projeto
	Gerência de Riscos	Gerência de Projeto
	Integração da Equipe	Gerência do Projeto
	Análise de Decisão e Resolução	Suporte
	Ambiente Organizacional para Integração	Suporte
Gerência Integrada de Fornecedores	Gerência de Projeto	
Nível 4 (Gerenciado Quantitativamente)	Desempenho do Processo Organizacional	Gerência de Processo
	Gerência Quantitativa do Projeto	Gerência de Projeto
Nível 5 (Em otimização)	Inovação Organizacional e Implantação	Gerência do Processo
	Análise e Resolução de Causas	Suporte

### **2.4.5. Modelo de Referência para Melhoria de Processos de Software (MR-MPS)**

Em dezembro de 2003, a SOFTEX<sup>5</sup>, lançou o projeto MPS-BR visando à melhoria do processo de software brasileiro com foco nas pequenas e médias empresas. O MPS.BR possui dois objetivos (WEBER et al., 2005): (1) desenvolver o MR-MPS, compatível com o CMMI e em conformidade com as normas ISO/IEC 12207 e 15504; e (2) implementar e avaliar o MR-MPS nas organizações brasileiras de software em todas as regiões do país.

Segundo Weber et al (2004), o diferencial do MPS.BR não está na definição de um novo modelo ou padrão, mas sim na estratégia de implementação criada de acordo com a realidade das organizações de software do Brasil. O MPS.BR também define regras para sua implementação e avaliação, dando sustentação e garantia de que está sendo empregado de forma coerente com as suas definições.

O MPS.BR é dividido em três componentes (SOFTEX, 2005): (i) Modelo de Referência de Melhoria de Processo de Software (MR-MPS), baseado nas normas NBR ISO/IEC 12207 e suas emendas 1 e 2, na norma ISO/IEC 15504 e compatível com o CMMI-SE/SW. O MR-MPS contém definições dos níveis de maturidade e da capacidade de processos; (ii) Processo e Método de Avaliação (MA-MPS), baseado na norma ISO/IEC 15504-2:2003, contém o processo e método de avaliação, os

<sup>5</sup> Associação para Promoção da Excelência do Software Brasileiro

requisitos para os avaliadores e os requisitos para averiguação da conformidade ao modelo MR-MPS; (iii) Modelo de Negócio (MN-MPS), que contém uma descrição das regras para a implementação do MR-MPS pelas empresas de consultoria, de software e de avaliação.

O Modelo de Referência MR-MPS define sete níveis de maturidade que são uma combinação entre processos e capacidade de processos, conforme a estrutura apresentada na Figura 2.7. Os níveis, seqüenciais e cumulativos, de maturidade do MR-MPS são: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado), sendo este o nível de maturidade inicial.

Weber et al. (2004) afirmam que um número maior de níveis de maturidade possibilita maior visibilidade dos resultados com prazos mais curtos.

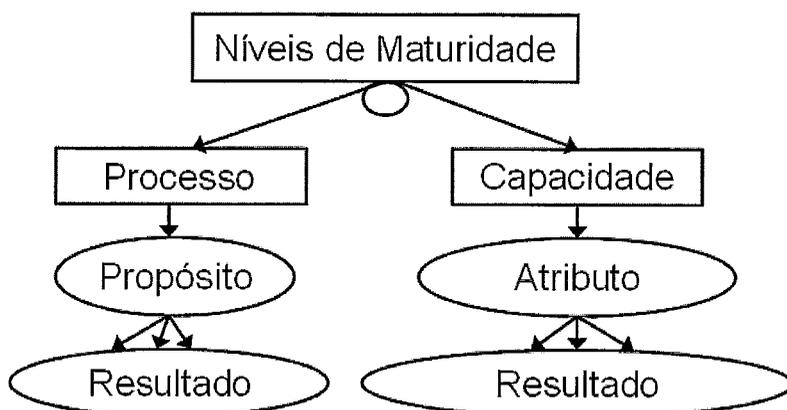


Figura 2.7 - Estrutura do MR-MPS (SOFTEX, 2005).

No MR-MPS o nível de capacidade do processo, assim como na ISO/IEC 15504-2, é dado pelo atendimento a um conjunto de atributos de processo. A capacidade estabelece o grau de refinamento e institucionalização com que o processo é executado na organização (SOFTEX, 2005).

Tabela 2.3 - Níveis de maturidade, processos e capacidade do MR-MPS (SOFTEX, 2005).

Nível	Processo	Atributos da Capacidade
A (mais alto)	Inovação e Implantação na Organização	Processo executado Processo gerenciado Produtos de trabalho gerenciados
	Análise e resolução de causas	Processo definido Processo implementado
B	Desempenho do Processo Organizacional	Processo executado Processo gerenciado Produtos de trabalho gerenciados
	Gerencia Quantitativa do Projeto	Processo definido Processo implementado
C	Análise de Decisão e Resolução	Processo executado Processo gerenciado Produtos de trabalho gerenciados
	Gerência de Riscos	Processo definido Processo implementado

Nível	Processo	Atributos da Capacidade
D	Desenvolvimento de Requisitos	Processo executado
	Solução Técnica	Processo gerenciado
	Integração do Produto	Produtos de trabalho gerenciados
	Instalação do Produto	Processo definido
	Liberação do Produto	Processo implementado
	Verificação	
	Validação	
E	Treinamento	Processo executado
	Avaliação e melhoria do Processo Organizacional	Processo gerenciado
	Adaptação do Processo para Gerência de Projeto	Produtos de trabalho gerenciados
F	Medição	Processo definido
	Gerência de Configuração	Processo implementado
	Aquisição	
	Garantia da Qualidade	
G	Gerência de Requisitos	Processo executado
	Gerência de Projeto	Processo gerenciado

## 2.5. Instrumentos para Avaliar a Efetividade das Melhorias em Processos de Software

Propostas de melhoria que não apresentam riscos de implantação e benefícios facilmente evidenciados podem ser imediatamente institucionalizadas. Outras, não provadas e/ou com características inovadoras, geralmente associadas a riscos e incertezas, precisam ser avaliadas quanto à sua efetividade.

As propostas de melhoria de processos de software resultam em ações voltadas para a implementação de alterações no processo. As alterações em processos de software, em geral, consistem em: incluir, alterar ou excluir atividades; alterar e incluir novos artefatos a serem gerados ao longo da execução do processo; alterar as diretrizes para execução das atividades; e adotar novas tecnologias (ferramentas, métodos, técnicas etc.) para apoiar a execução das atividades.

A seguir, serão apresentados instrumentos que auxiliam na avaliação da efetividade das propostas de melhoria em processos de software.

### 2.5.1. Estudos experimentais

“Com o objetivo de melhorar seus processos de desenvolvimento de software, as organizações frequentemente adotam novas tecnologias sem evidências relacionadas à efetividade em detrimento de outras que possivelmente poderiam ser mais úteis” (ZELKOWITZ; WALLACE; BINKLEY, 2003).

O fato do desenvolvimento de software ser um processo social incrementa a complexidade na realização de estudos experimentais para analisar a efetividade de novas tecnologias. Processos de natureza não-determinística apresentam um grande número de variáveis que precisam ser observadas. Assim, recomenda-se caracterizar

o contexto do problema através da observação de variáveis e do registro de possíveis explicações para os comportamentos observados; testar as hipóteses na prática, registrando probabilidades, novos valores e grau de importância de variáveis, incluindo novas variáveis e excluindo outras; e, por fim, realizar análises com o objetivo de reduzir incertezas (PFLEEGER, 1999).

Dependendo do propósito da análise e das condições para a investigação, três principais tipos de estratégias podem ser aplicadas: *Surveys*, Estudos de Caso e Experimentos (WOHLIN et al., 2000).

#### 2.5.1.1. *Surveys*

*Survey* é um instrumento abrangente para coletar informação, descrever, comparar ou elucidar conhecimento, atitudes e comportamento (PFLEEGER; KITCHENHAM, 2001).

Geralmente o objetivo do *survey* é descrever um fenômeno de interesse ou avaliar o impacto de alguma intervenção (KITCHENHAM; PFLEEGER, 2002). No primeiro caso, o intuito é determinar a distribuição de certas características ou atributos. A preocupação não é o porquê da existência de uma determinada distribuição, mas sim identificar qual é a distribuição (WOHLIN et al., 2000, p.11). No segundo caso, os *surveys* são úteis para apoiar a avaliação dos efeitos de tecnologias antes de serem utilizadas (potenciais efeitos) ou quando já estão sendo utilizadas no ambiente organizacional.

#### 2.5.1.2. Estudos de Caso

Em estudos de caso, projetos são monitorados e dados são coletados durante a sua execução. Com uma pequena adição nos custos do projeto, valiosas informações podem ser obtidas (ZELKOWITZ; WALLACE, 1998). Kitchenham et al. (1995) defendem que estudos de caso auxiliam a indústria a avaliar os benefícios de métodos e ferramentas, além de prover uma forma econômica e efetiva para assegurar que a mudança no processo propiciará os resultados desejados.

Um aspecto negativo é a dificuldade para generalizar os resultados (WOHLIN et al., 2000), determinar tendências e prover uma validade estatística devido à especificidade de cada projeto (ZELKOWITZ; WALLACE, 1998).

Contudo, Kitchenham et al. (1995) provêem orientações que permitem identificar em quais condições os estudos de caso são mais adequados e fornecem um guia para auxiliar a condução destes estudos de forma a prover mais rigor no decorrer da investigação. As orientações na forma de um guia consistem na descrição de sete

passos para a realização de estudos de caso, são eles: (1) Definir hipóteses; (2) Selecionar projetos-piloto; (3) Identificar o método de comparação; (4) Minimizar os efeitos dos fatores de confusão; (5) Planejar o estudo de caso; (6) Monitorar o estudo de caso conforme o plano; e (7) Analisar e reportar os resultados.

### 2.5.1.3. Experimentos

Um experimento é uma investigação formal, rigorosa e controlada (WOHLIN et al., 2000). Este tipo de estudo normalmente é realizado em ambientes que provêm alto nível de controle e que permitem a manipulação de uma ou mais variáveis e o controle de outras, conforme o interesse do estudo. Durante o experimento, os efeitos da manipulação são mensurados, e ao final, os dados coletados podem ser submetidos a análises para determinar o nível de significância estatística obtido.

A Figura 2.8 ilustra os principais elementos envolvidos em um experimento utilizando a terminologia adotada por Wohlin et al. (2000). O objeto de estudo possui os atributos **A**, **B** e **C**. Os atributos considerados relevantes para o estudo, isto é, que serão controlados durante o experimento são denominados variáveis independentes. Os atributos manipulados para que os efeitos possam ser observados no decorrer do estudo, são denominados fatores. Na Figura 2.8, **B** é um fator. A manipulação de **B** é realizada através de tratamentos que, neste caso, consistem na atribuição dos elementos **2**, **4** e **7** ao fator **B**. Para cada tratamento, o objeto sob análise é submetido ao ambiente, interage com os participantes e sofre influência das variáveis ambientais (**X**, **Y**, **Z**). Outros fatores não identificados a priori também podem exercer alguma influência nos efeitos obtidos (**fatores de confusão**). Durante a interação do objeto com o ambiente, os dados de interesse são coletados e atribuídos às variáveis dependentes, ou seja, às variáveis de resposta, responsáveis por representar os efeitos obtidos com a experiência.

Vale ressaltar que o objetivo do experimento é coletar um montante representativo de dados para que seja possível obter um resultado estatisticamente significativo (ZELKOWITZ et al., 2003).

Segundo Sjøberg et al. (2005), a condução de experimentos controlados é o método clássico para identificar os relacionamentos causa-efeito. Porém, Briand et al. (1995) alegam que a realização de experimentos não é adequada em ambiente industrial e fornecem dois argumentos: a difusão do tratamento devido à ausência de controle sobre a comunicação e a inconveniente necessidade de reter o tratamento em função de ações voluntárias de pessoas que desejam participar dos estudos. Wohlin et al. (2000, p. 14) reforçam estes argumentos afirmando que a impossibilidade de

controlar alguns fatores no ambiente industrial dificulta a realização de experimentos em uma situação *on-line*.

Ciente da necessidade de convencer a indústria sobre a validade e aplicabilidade dos resultados obtidos a partir de estudos experimentais, Sjøberg et al. (2002) ressaltam dois obstáculos: a ausência de controle nas experiências em ambiente industrial e a ausência de realismo na realização de experimentos controlados. Fuggetta (2000) também enfatiza a necessidade de assegurar que o estudo foi conduzido corretamente (validade interna) e, além disso, sobre a necessidade de entender sob quais circunstâncias os resultados do estudo podem ser aplicados em diferentes contextos (validade externa).

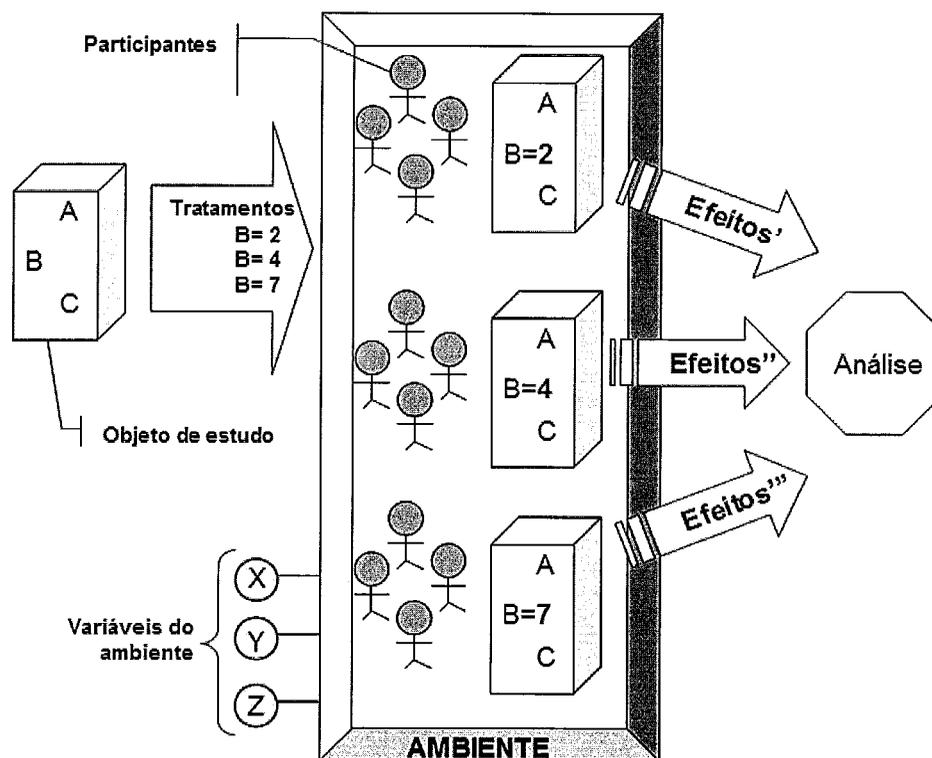


Figura 2.8 - Experimento: um fator e três tratamentos

### 2.5.2. Simulação

A Simulação é um método controlado que se apresenta como alternativa ao uso de pilotos para avaliar os efeitos das mudanças e as relações entre as variáveis que estão sendo observadas (PFAHL; BIRK, 2000; TVEDT; COLLOFELLO, 1995).

Uma das principais motivações para desenvolver ou utilizar um modelo de simulação reside no baixo custo, na limitação de riscos e/ou no decremento do esforço logístico para manipular o sistema de interesse (KELLNER; MADACHY; RAFFO, 1999). Neste sentido, diversas iniciativas têm sido publicadas para demonstrar a aplicabilidade da simulação na melhoria de processos.

Höst et al. (2001) utiliza a simulação de um processo de engenharia de requisitos para observar as condições de sobrecarga e identificar mudanças que possam remover os gargalos do processo. Pfahl e Lebsanft (2000) também utilizam simulação no processo de engenharia de requisitos para analisar qual o esforço financeiro necessário para estabilizar e alcançar o ponto "ótimo" do processo.

Raffo (1993) apresenta um exemplo onde potenciais mudanças no processo são quantitativamente avaliadas através de uma abordagem de simulação estocástica, antes de comprometer recursos e esforço para a implantação das mudanças. Além de tentar prever o impacto da mudança realizada nos processos, esta proposta visa apoiar a priorização de ações de melhoria e avaliar o desempenho do modelo de processo atual com relação ao modelo planejado.

Ahmed-Nacer (2004) descreve uma abordagem para modificar modelos de processo e processos de software que suporta diferentes estratégias de avaliação e permite a adaptação do processo "em vôo" (durante a execução). Durante a simulação, o modelo de processo de software é atualizado e validado através de múltiplas simulações antes de sua reexecução. A abordagem não considera o fator humano, bem como as diversas variáveis ambientais que podem influenciar a avaliação da mudança.

Para modelar sistemas dinâmicos é necessária uma base de conhecimento que possua relações pré-estabelecidas entre fatores que influenciam o comportamento do modelo. Kellner et al. (1999) apontam algumas questões não triviais que precisam ser tratadas ao optar pelo desenvolvimento/utilização de modelos, a exemplo da identificação das variáveis de resposta e a definição dos parâmetros de entrada, que podem chegar a algumas centenas.

Zelkowitz et al. (2003) afirma que a limitação da simulação reside na acurácia da representação dos modelos, ou seja, a simulação é conduzida em um ambiente artificial controlado que pode não apresentar todas as características que exercem influência nas atividades em um ambiente real. Esta limitação dificulta a observação dos efeitos reais das mudanças no ambiente industrial. Portanto, a simulação é útil para observar efeitos considerando um contexto limitado, levando em consideração que o mundo real é muito mais rico e exerce muitas influências sobre os projetos. Münch e Armbrust (2003) apresentam uma breve, porém valiosa discussão das limitações encontradas durante a experiência da construção de um modelo para simulação de processos de software.

Neste sentido, instrumentos como IMMoS<sup>6</sup> (PFAHL; RUHE, 2002) e VSEL<sup>7</sup> (MÜNCH; PFAHL; RUS, 2005) têm sido evoluídos para suprimir óbices relacionados ao custo para construção dos modelos, a acurácia da representação e facilidades na interação com o usuário.

### 2.5.3. Benchmarking

No *benchmarking* a avaliação das propostas de melhoria se dá através da observação externa de como as ações de melhoria são percebidas. Ou seja, observam-se outras organizações que realizam as práticas propostas como melhoria para a organização e verifica-se o resultado obtido por elas a partir do uso destas práticas.

O principal problema desta abordagem é o contexto organizacional, pois o sucesso da adoção de uma prática ou tecnologia em determinadas organizações não garante o mesmo sucesso em outras. Além disso, é importante identificar o que faz determinadas práticas passíveis de comparação.

Fogle et al. (2001) afirmam que, para executar *benchmarking*, os envolvidos devem possuir conhecimento consolidado no domínio a ser observado, entender o processo de *benchmarking* e possuir habilidades para organizar, analisar e compreender os dados. Os autores também recomendam o uso de metodologias comprovadas para não incorrer no desperdício de recursos e obter resultados úteis para a organização.

Beitz e Wieczorek (2000) ilustram como utilizar *benchmarking* no contexto de programas de melhoria e demonstram a aplicação da técnica OSR (*Optimised Set Reduction*) para viabilizar a comparação dos dados coletados a partir outras organizações.

### 2.5.4. Pilotos

A execução de projetos-piloto pode servir a vários propósitos, dentre eles: (i) Identificar questões relacionadas ao conhecimento tácito (SHULL et al., 2004), tornando-o explícito à medida que o piloto é executado, servindo como uma importante fonte de lições aprendidas e agregando valor à organização (BRIAND et al., 1995); (ii) Avaliar a conformidade do processo antes que qualquer esforço significativo de replicação seja realizado (SHULL et al., 2004); (iii) Avaliar mudanças significativas ou

<sup>6</sup> Integrated Measurement, Modeling, and Simulation. Método para mensuração, modelagem e simulação integrada de processos de software.

<sup>7</sup> Virtual Software Engineering Laboratory. Consiste em um conjunto de módulos de simulação de processos de software que podem apoiar a solução de um problema, prover suporte à decisão e/ou analisar um conjunto de atividades dentro de um contexto específico.

melhorias inovadoras, que não foram experimentadas e possuem alto risco, antes de serem amplamente disseminadas na organização (CHRISISS et al., 2003); e (iv) obter um retorno a respeito das adaptações necessárias às novas tecnologias para uso na organização (BRIAND et al., 1995).

Fowler e Rifkin (1990) afirmam que a implantação de novos procedimentos e tecnologias que apóiam a melhoria de processos é uma atividade contínua e de longo prazo e que estas implantações devem ser iniciadas com a realização de um piloto. “Pilotos são essenciais quando a organização não tem experiência com a tecnologia ou a tecnologia será aplicada em um novo domínio com uma equipe inexperiente.” (Ibidem, p. 51, tradução nossa). Fowler e Rifkin (1990) tratam a realização de pilotos como experimentos controlados e sugerem um conjunto de procedimentos para a realização de pilotos que envolvem a utilização de novas tecnologias:

**Obter a tecnologia.** O grupo de processo da organização deve selecionar o método ou a ferramenta dentre as disponíveis, contatar os fornecedores (internos ou externos) e providenciar que esta fique disponível na organização;

**Formar especialistas na organização.** Prover um treinamento específico para os profissionais que apoiarão a realização do piloto e que, posteriormente, serão encarregados de elaborar os guias de apoio e disseminar a tecnologia para toda a organização. O objetivo é adquirir conhecimento e mantê-lo disponível na organização;

**Elaborar guias para uso da tecnologia.** Na realização de um piloto o treinamento pode não ser necessário, contudo, guias que orientem o uso da tecnologia no contexto do piloto devem ser fornecidos;

**Adaptar o treinamento às necessidades da equipe do projeto-piloto.** Se o treinamento for necessário para os engenheiros que participam do projeto-piloto, ele deve ser adequado e apresentar coerência com os guias;

**Preparar a tecnologia para uso.** Além do treinamento, a tecnologia deve ser configurada sob medida para atender às necessidades do projeto;

**Prover o treinamento para uso da tecnologia.** O treinamento a todos os envolvidos deve ser realizado imediatamente antes do uso da tecnologia. Prover o treinamento com alguma antecedência pode exercer uma influência negativa ao longo da execução do projeto. O treinamento também deve ser utilizado para apresentar o problema a ser atacado, construir o comprometimento da equipe e incentivá-la a contribuir com iniciativa;

**Iniciar o uso da tecnologia.** Ao iniciar o uso, durante as primeiras semanas, o grupo de processo deve dar uma atenção especial para capturar as dificuldades encontradas e promover ajustes nos guias, treinamento e na configuração da tecnologia. É necessário distinguir os problemas da tecnologia dos problemas em decorrência do mau uso da mesma;

**Prover suporte contínuo.** Suporte contínuo deve ser fornecido para apoiar a equipe do projeto-piloto e ao mesmo tempo coletar a opinião dos participantes durante o projeto.

Outro argumento em favor do uso de pilotos para introdução de novas tecnologias em processos de software é que, mesmo que existam evidências comprovando resultados positivos, é prudente avaliar se a organização também será contemplada com estes benefícios (BRIAND et al., 1995). O Laboratório de Engenharia de Software da NASA (SEL), por exemplo, só adota novos processos após a realização de um ou mais projetos-piloto para verificar a efetividade no ambiente e adequá-los para uso (ZELKOWITZ, 1996; BASILI et al., 2002).

O uso de pilotos também pode apoiar de forma significativa na diminuição do tempo para institucionalização das melhorias, a exemplo da empresa Silicon and Software Systems, que reduziu o tempo médio para avaliar e introduzir mudanças nos processos de 533 dias para menos de 112 dias através do uso gerenciado de projetos-piloto (O'HARA, 2000).

A execução de pilotos também pode diminuir o planejamento para implantar a mudança. Quando o piloto é realizado de forma planejada, as necessidades de treinamento e a dimensão do esforço e recursos requeridos são percebidas. Estes dados podem auxiliar na elaboração da estratégia de implantação e no cômputo das estimativas que integram o planejamento para a mudança.

Os resultados negativos de um projeto-piloto planejado e executado adequadamente devem ser encarados de uma forma positiva. O principal ganho nestes casos é a identificação de uma "pseudo melhoria" que não traria benefícios e consumiria esforço e investimento para ser implantada na organização. Além disso, a experiência provê um valioso conhecimento a respeito dos aspectos observados no piloto.

### **2.5.5. Combinação de abordagens**

Para cada tipo de iniciativa de melhoria existem métodos que podem ser combinados para avaliar a efetividade das mudanças propostas.

Por exemplo, utilizar um projeto-piloto para avaliar um processo de desenvolvimento completamente novo, sem antes avaliar as chances de sucesso é perigoso. Nessas ocasiões, antes do piloto, é recomendado combinar *benchmarking*, entrevistas e avaliações de especialistas para evitar desgastes desnecessários.

Na simplificação de processos (DANTEKAR; PERRY; VOTTA, 1997) a simulação é um mecanismo bastante adequado para avaliar os riscos. Neste caso, uma das utilidades da simulação seria priorizar as propostas de melhoria, ou seja, identificar aquelas que possibilitam melhorias mais significativas e com menos riscos e, em seguida, avaliar a proposta com o piloto.

Na melhoria contínua, mais baseada em passos pequenos e evolutivos do que em inovações revolucionárias (PAULK et al., 1993b), a aplicação de *surveys* antes da realização do piloto pode ser útil para identificar elementos que possam influenciar a avaliação.

## **2.6. Modelos em Prol da Qualidade de Software e o Uso de Projetos-Piloto**

Modelos que tratam da melhoria de processos de software, tais como IDEAL (MCFEELEY, 1996), QIP (BASILI et al., 1994), CMMI (CHRISISS et al. 2003), MR-MPS (SOFTEX, 2005) e a ISO/IEC 15504 (ISO/IEC, 2003), sugerem o uso de projetos-piloto para avaliar a efetividade das ações em prol da melhoria quando há riscos e incertezas associadas.

O IDEAL (MCFEELEY, 1996) sugere que o uso de pilotos seja na fase de ação, quando há necessidade de avaliar potenciais soluções de melhoria.

No QIP (BASILI et al., 1994) um projeto é utilizado como piloto no primeiro ciclo (controlado) durante a fase de execução e a partir dele são coletados indicadores para prevenir e resolver os problemas observados.

O CMMI (CHRISISS et al. 2003) recomenda o uso de pilotos a partir do nível 3, nas áreas Foco no Processo Organizacional e Análise de Decisão e Resolução. Na área Foco no Processo Organizacional, mudanças importantes devem ser testadas por meio de pilotos antes de serem institucionalizadas. Em Análise de Decisão e Resolução, o piloto serve de instrumento para revelar problemas, avaliar diferentes critérios e alternativas que devem ser consideradas para anular um potencial viés na decisão. No nível 5, na área de Inovação e Implantação na Organização, pilotos são conduzidos para avaliar mudanças significativas envolvendo melhorias não testadas e de alto risco ou melhorias inovadoras, antes que elas sejam amplamente difundidas (CHRISISS et al., 2003). Essa área de processo também ratifica a necessidade de

planejar o piloto, obter o comprometimento dos envolvidos, executá-lo em um ambiente comum aos demais projetos da organização, acompanhar a sua execução, verificá-lo com relação ao que foi planejado, revisar e documentar os seus resultados (melhorias, problemas encontrados e lições aprendidas).

O MR-MPS (SOFTEX, 2005) faz referência ao uso de pilotos no nível A para apoiar a tomada de decisão quanto à institucionalização das ações de melhoria. O uso de pilotos auxilia a análise e priorização das propostas de melhoria do processo e de melhoria tecnológica. O modelo afirma que os projetos-pilotos devem ser planejados e os resultados e lições aprendidas registradas.

A ISO/IEC 15504 (ISO/IEC, 2003) menciona o uso de pilotos no contexto dos programas de melhoria, na avaliação das propostas obtidas a partir da análise das oportunidades de melhoria. A norma preconiza que o programa seja conduzido como um projeto que deve ser planejado, mensurado, monitorado e ter seus resultados avaliados.

A partir da menção feita ao termo “piloto” pelos modelos supracitados, observa-se uma visão compartilhada do conceito no contexto da melhoria de processos de software que utilizaremos neste trabalho: piloto é um instrumento utilizado para avaliar os efeitos da mudança em processos de software em um ambiente real.

Em alguns modelos, a maneira que as práticas são sugeridas não é explícita, portanto dependem da interpretação do leitor. A ISO/IEC 15504 (ISO/IEC, 2003), por exemplo, associa o estabelecimento de métricas, planejamento e treinamento ao projeto do programa de melhoria como um todo, não especificamente para a realização de um piloto, que pode ou não estar contido no programa de melhoria. Portanto, subentende-se que todo piloto contido em um programa de melhoria deverá ser submetido a tais práticas.

Mesmo sugerindo práticas e recomendando o uso de pilotos em iniciativas de melhoria de processos, estes modelos não descrevem procedimentos que permitam avaliar de maneira consistente os efeitos obtidos a partir das mudanças dos processos.

## **2.7. Considerações Finais**

As precondições para empreender de forma efetiva a melhoria de processos de software são: possuir processos definidos, mensurá-los e avaliá-los de forma contínua e periódica.

A adoção de modelos de qualidade como referencial pode auxiliar as organizações que almejam atingir altos níveis de maturidade e ao mesmo tempo prover credibilidade ao mercado através dos mecanismos de avaliação, como as promovidas pelo SEI<sup>8</sup> e pela SOFTEX<sup>9</sup>.

Os modelos e normas em prol da qualidade sugerem várias práticas que podem ser utilizadas para avaliar a efetividade das propostas de melhoria, antes que estas sejam institucionalizadas. Porém, não descrevem como estas práticas devem ser conduzidas para prover avaliações com razoável nível de acurácia.

Estudos experimentais, simulação, *benchmarking* e a realização de pilotos são mecanismos que podem ser utilizados para avaliar a efetividade das propostas de melhoria, inclusive, podem ser combinados para potencializar os resultados da avaliação. Porém: (i) Para a realização de estudos experimentais é requerido conhecimento em experimentação e experiência para determinar qual a estratégia mais adequada para realizar o estudo; (ii) A simulação requer modelos acurados, cuja elaboração não é trivial e envolve um esforço e uma quantidade de conhecimento que dificilmente pequenas e médias empresas teriam condições de patrocinar; (iii) *Benchmarking* não são apropriados para avaliar propostas que oferecem um nível de risco elevado, dado o grau de incerteza que ele provê; (iv) Para o uso adequado de pilotos é necessário planejamento e monitoração para que seja possível estender ao máximo os efeitos observados aos demais projetos da organização.

O próximo capítulo apresenta a contribuição dos ambientes de engenharia de software no que tange a melhoria contínua de processos de software. Neste contexto, um breve histórico sobre a Estação TABA e sua estratégia para avaliação e melhoria em níveis também serão apresentados.

---

<sup>8</sup> Software Engineering Institute, instituto vinculado à Universidade Carnegie Mellon.

<sup>9</sup> Sociedade para Promoção da Excelência do Software Brasileiro, organização da sociedade civil de interesse público que atua de diversas formas, e em diversas frentes, para o crescimento e difusão da indústria brasileira de software.

## **Capítulo 3. Ambientes de Engenharia de Software Centrados em Processo e a Estação TABA**

---

### **3.1. Introdução**

Na opinião de Sharon e Anderson (1997), um ambiente completo de engenharia de software é composto por todas as atividades para desenvolvimento, reengenharia e manutenção de sistemas de software. E essas atividades podem ser automatizadas por ferramentas, procedimentos manuais ou ambos.

Os primeiros esforços para a construção de ambientes de apoio à engenharia de software, segundo Harrison et al. (2000), consistiram na integração de ferramentas simples, como editores, compiladores e depuradores. Para Harrison et al. (2000) o primeiro esforço significativo na produção de ambientes integrados de desenvolvimento teve o propósito de produzir ambientes de apoio à programação, apoiando especificamente atividades de codificação e excluindo as demais atividades, tais como engenharia de requisitos, especificação, projeto, teste e análise. O autor afirma que a difusão dos modelos de ciclo de vida de software permitiu a observação da conexão existente entre todas as atividades de engenharia de software e como elas se influenciam mutuamente. Este fato fomentou a necessidade de apoiar estas atividades de forma integrada através do ciclo de vida do software e representou o ponto de partida para a pesquisa em ambientes de engenharia de software (AES).

Balzer e Gruhn (2001) apresentam vantagens e desvantagens para AES que oferecem suporte a apenas um tipo de processo. Dentre as vantagens, a possibilidade de assegurar um alto nível de consistência entre os documentos produzidos, o controle detalhado do estado do processo e certo grau de automação das atividades que compõem o processo. Como grande desvantagem, Balzer e Gruhn (2001) destacam a inconveniência da necessidade de utilizar ferramentas não integradas para desenvolver software.

Para suplantiar esse e outros óbices, pesquisas têm sido realizadas para desenvolver ambientes de engenharia de software centrados em processos que possam oferecer apoio à execução de mais de um tipo de processo de software e que contribuam efetivamente com a melhoria contínua da organização.

Este capítulo apresenta uma visão geral sobre ambientes de engenharia de software centrados em processos, discorre sobre a Estação TABA e comenta como

esses ambientes podem apoiar a avaliação da efetividade das propostas de melhorias de processo.

### **3.2. Ambientes de Engenharia de Software Centrados em Processos (AESCPs)**

Os AESCPs são ambientes que apóiam a criação e a utilização de modelos de processo de software (FUGGETTA, 2000). Estes ambientes provêm uma representação explícita do modelo do processo que especifica como executar as atividades de desenvolvimento de software, os papéis e tarefas dos desenvolvedores e como utilizar e controlar as ferramentas de desenvolvimento (AMBRIOLA et al., 1997). Ellmer (1995) defende que um AESCP tem que ter "consciência" sobre o tipo de processo, isto é, da estrutura e comportamento dos processos de engenharia de software, bem como das unidades organizacionais que executam estes processos para apoiar adequadamente os projetos de software.

Arbaoui et al. (2002) enumeram quatro requisitos desejáveis para AESCPs: (i) prover representação e suporte à execução do processo através de uma linguagem sem ambigüidades e inteligível; (ii) permitir a ordenação dinâmica das atividades do processo de software; (iii) apoiar a execução de processos de software de forma distribuída; e (iv) apoiar a evolução do processo de software. Além disso, segundo Fuggeta (2000), um AESCP deve: integrar-se suavemente ao "tradicional" ambiente de desenvolvimento, complementando-o de forma não intrusiva; ser flexível a ponto de atender a natureza criativa das atividades de desenvolvimento; e deve prover ao engenheiro de software uma clara visão do processo de desenvolvimento de software.

Dezenas de exemplos de AESCPs podem ser encontrados na literatura, Balzer et al. (2001) listam mais de uma dúzia. Estudos comparativos e de caracterização destes ambientes podem ser encontrados em Garg e Jazayeri (1996), Ambriola et al. (1997) e Arbaoui et al. (2002).

Com relação à contribuição dos AESCPs à melhoria de processos de software, Ellmer (1995) realizou uma investigação que consistiu na adoção do CMM como modelo de referência e na construção de um protótipo AESCP para apoiar a avaliação. Embora as conclusões do estudo apresentem um alto fator intuitivo, é possível observar que AESCPs bem projetados podem contribuir com o aumento da maturidade dos processos nas organizações.

### **3.3. A Estação TABA**

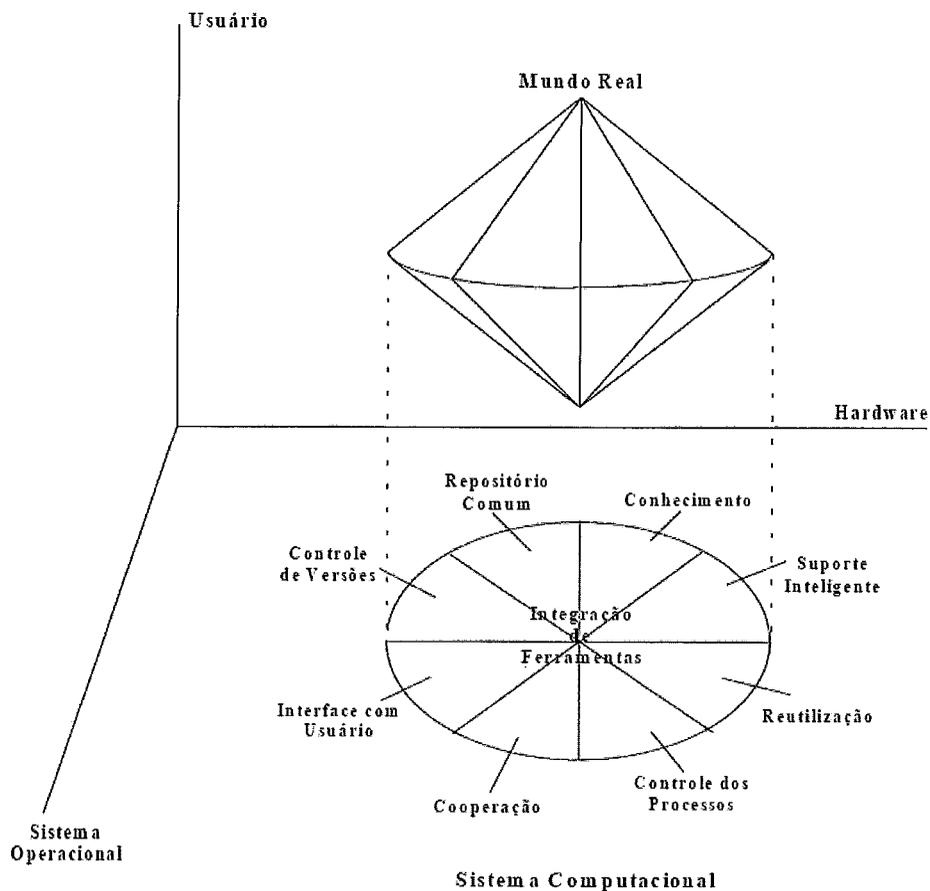
A Estação TABA é um ambiente de desenvolvimento de software desenvolvido pela COPPE/UFRJ que apóia a definição e execução de processos de software. Sua utilização no âmbito industrial público e privado, como instrumento de apoio a programas de melhoria, têm colecionado casos de sucesso, a exemplo dos resultados positivos obtidos nas avaliações SCAMPI (CMU/SEI, 2001) na empresa Relacional (DUARTE et al., 2005) e no Tribunal Superior Eleitoral (MONTONI et al., 2005).

#### **3.3.1. Breve histórico**

O marco da concepção da Estação TABA foi a publicação “Taba: A Heuristic Workstation for Software Development”, que a definiu como um meta-ambiente capaz de gerar, através de instanciação, ambientes de desenvolvimento de software (ADS) adequados às particularidades de processos de desenvolvimento e de projetos específicos (ROCHA et al., 1990). Neste contexto, um meta-ambiente foi definido como um ambiente que abriga um conjunto de programas que interagem com os usuários para definir interfaces, selecionar ferramentas e estabelecer os tipos de objetos que irão compor o ambiente de desenvolvimento específico.

Nesses 16 anos de evolução, a Estação TABA vem sendo estendida e contemplada com novas funcionalidades.

A primeira versão da Estação TABA foi implementada em Eiffel e foi definida por Travassos (1994) como um conjunto de componentes integrados que possuem controle sobre sua existência, suas informações, seus estados e suas funcionalidades básicas associadas (Figura 3.1). O uso conjunto destes componentes também definiu a filosofia de integração da Estação TABA que consiste em: (i) integração de apresentação e interação, que possibilita a homogeneização das formas de apresentação das informações e das técnicas de interação com o usuário; (ii) integração de gerenciamento e controle, que provê, para ferramentas e para o ambiente, serviços e funcionalidades que permitam o funcionamento de forma organizada ao longo do processo de desenvolvimento; (iii) integração de dados, que estabelece a forma como as ferramentas da Estação realizarão o tratamento das informações; e (iv) integração do conhecimento, que torna possível os serviços básicos de armazenamento, gerenciamento e utilização do conhecimento descrito e adquirido.



**Figura 3.1 - Estrutura da Estação TABA definida por Travassos (1994)**

No ano seguinte, o primeiro ambiente instanciado foi implementado utilizando o modelo de integração da Estação TABA para apoiar o desenvolvimento de sistemas baseados em conhecimento (WERNECK, 1995).

Posteriormente, Falbo (1998) agregou a idéia do uso de servidores de conhecimento para prover componentes, construídos a partir de ontologias, que pudessem ser utilizados por várias ferramentas.

Em 1999, a Estação TABA foi estendida para ser capaz de instanciar ambientes de desenvolvimento de software orientados a domínio (ADSOD), que habilitou o ambiente a fornecer conhecimento relacionado com o domínio do software que está sendo desenvolvido ou mantido (OLIVEIRA, 1999).

Em seguida, o ambiente tornou-se apto a instanciar ambientes de desenvolvimento de software orientados à organização (ADSOrg) (VILLELA, 2004), definidos como sendo ADS que apóiam a gerência do conhecimento ao longo dos processos de desenvolvimento e manutenção de software no âmbito organizacional.

Atualmente a Estação TABA reúne 32 ferramentas, que oferecem suporte às atividades dos processos de desenvolvimento e manutenção de software, que vão desde o apoio à instanciação de processos, atividades de planejamento, mensuração

e gerência de configuração até avaliação *post-mortem* e apoio à captura do conhecimento, contemplando grande parte das atividades presentes no ciclo de vida do software.

### 3.3.2. Visão geral da abordagem para definição de processos

Oliveira (1999) descreve o modelo para definição de processos utilizado pela Estação TABA (Figura 3.2), que consiste em três etapas: definição do processo padrão, especialização do processo padrão e instanciação dos processos para os projetos. A partir da combinação de modelos e normas de qualidade, melhores práticas em engenharia de software e no exame das características do desenvolvimento de software na organização, é concebido um modelo de processo padrão adequado a cada organização. A partir do modelo de processo padrão, modelos de processos especializados são definidos de acordo com as abordagens de desenvolvimento adotadas na organização (paradigmas, métodos de desenvolvimento e tipos de software, por exemplo). Por fim, os processos da organização são adaptados e instanciados considerando as características e particularidades de cada projeto.

O processo padrão e os processos especializados da organização integram o ambiente configurado da Estação TABA, que é fornecido para cada organização de forma personalizada. Para cada projeto da organização, através da interação com o ambiente configurado da Estação TABA, é criado um ambiente instanciado contendo um processo instanciado e devidamente adaptado que atende às especificidades do projeto.

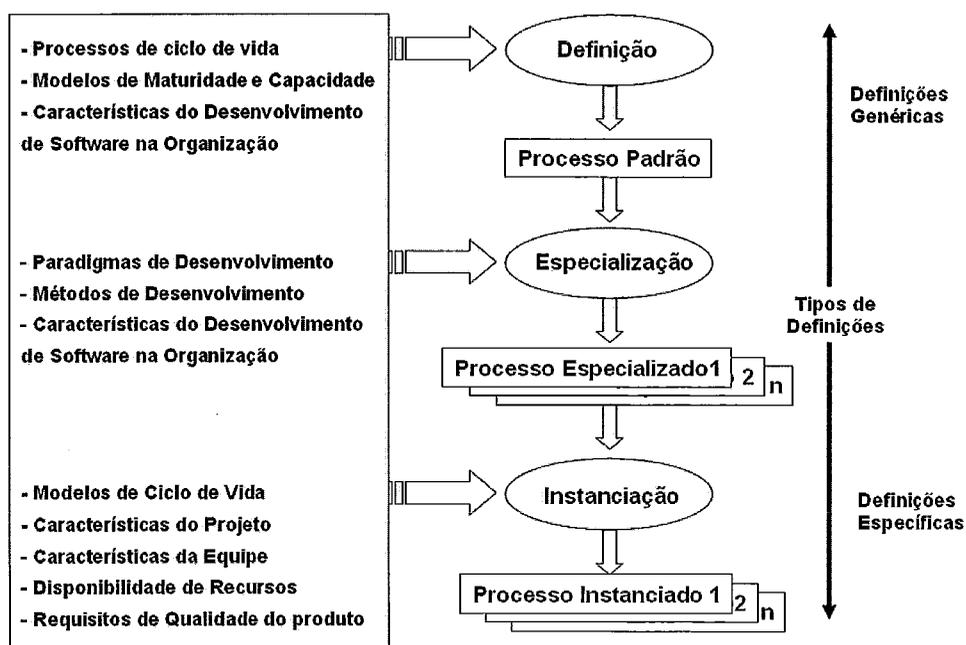


Figura 3.2 - Abordagem para definição de processos na Estação TABA (VILLELA et al., 2004).

### 3.3.3. Estratégia de avaliação e melhoria em níveis da Estação TABA

Incluída nos recentes avanços da Estação TABA, está a Estratégia de Avaliação e Melhoria em Níveis (CAMPOS et al., 2005) (Figura 3.3), compatível com o modelo para a definição de processos (Figura 3.2).

Na Estratégia de Avaliação e Melhoria e Níveis, em um primeiro momento, as avaliações dos processos nos projetos (processos instanciados) são realizadas a partir do relatório de aderência aos processos, das medidas coletadas, das observações feitas pela equipe do projeto e da avaliação *post-mortem*, sempre observando o escopo de cada projeto.

Em uma segunda etapa, a avaliação dos processos da organização é feita considerando os relatórios de avaliação dos processos instanciados, as características dos processos especializados e as características do processo padrão que foram aplicadas em todos os projetos desenvolvidos.

A partir das avaliações realizadas nos processos da organização, são identificadas oportunidades de melhorias. Estas oportunidades são analisadas e dão origem às propostas de melhoria, que são avaliadas e priorizadas.

Na avaliação das propostas de melhoria são considerados fatores como risco, impacto esperado, esforço necessário, dentre outros aspectos. O resultado das avaliações das propostas de melhoria pode sugerir a realização de um **piloto** para que os efeitos das mudanças sejam conhecidos e devidamente mensurados. A experiência a partir da realização de pilotos, também viabiliza a observação de aspectos que podem ser aperfeiçoados para que a melhoria seja institucionalizada da melhor maneira possível.

Concluídas as avaliações das propostas de melhoria, aquelas que demonstraram efetividade são institucionalizadas, ou seja, passam a integrar os processos da organização e passam a ser utilizadas por toda a organização.

Transcendendo o escopo das organizações, a estratégia visa atuar em um terceiro nível (multi-organizacional), onde as avaliações e melhorias realizadas em cada organização são analisadas, podendo resultar em melhorias no processo de definição de processo para as próprias organizações e na base de ativos compartilhada por todas as organizações.

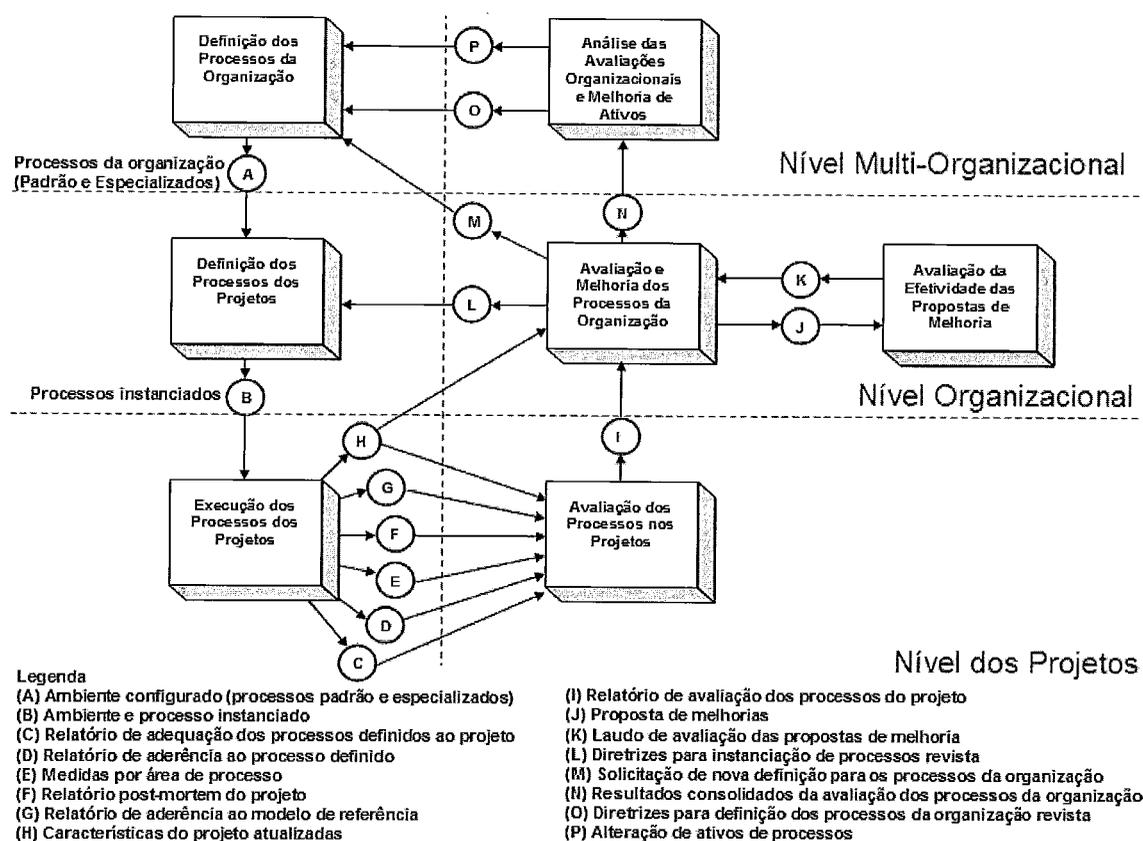


Figura 3.3 - Estratégia para avaliação e melhoria de processos em níveis (CAMPOS, 2005).

### 3.4. AESCP e a Avaliação da Efetividade das Propostas de Melhoria

Durante as iniciativas de melhoria, organizações analisam os pontos fortes e fracos, identificam oportunidades de melhoria e confeccionam propostas que facilitem o alcance dos objetivos e metas estipuladas pela organização.

Neste cenário, AESCPs provêm facilidades no que tange a execução e mensuração das atividades, pois o uso destes ambientes tende a diminuir a variabilidade da execução dos processos e, em geral, dispõem de instrumentos que permitem monitorar as atividades do processo (AMBRIOLA et al., 2002).

Na indústria de software, que possui um baixo nível de controle para estudos experimentais, estas facilidades providas pelos AESCPs imprimem uma condição favorável à avaliação dos efeitos causados pelas mudanças em processos de software em prol da melhoria.

Assim, a disponibilidade de um AESCP e a possibilidade de integrá-lo ao instrumento de avaliação das propostas de melhoria é um requisito desejável.

### **3.5. Considerações Finais**

Tendo em vista os conceitos apresentados por Fuggetta (2000), Ambriola et al. (1997), Ellmer (1995) e os requisitos apresentados por Arbaoui et al. (2002) para ambientes de engenharia de software orientados a processos, é possível afirmar que a Estação TABA é um ambiente de engenharia de software centrado em processos. Além disso, dadas as recentes evoluções da Estação TABA, também se pode afirmar que atualmente a Estação TABA é um ambiente de engenharia de software orientado a melhoria, conforme a caracterização apresentada por Basili e Rombach (1988) para tais ambientes.

Diante das facilidades providas por AESCPs, será apresentada nesta dissertação, a aplicação da abordagem para realização de pilotos na Estação TABA. Dentre os critérios utilizados para seleção deste ambiente de engenharia de software, destacamos: (i) ampla utilização na indústria, (ii) maturidade, e (iii) conveniência, especialmente com relação à proximidade geográfica dos usuários e da equipe de desenvolvimento do ambiente.

Dado o contexto, o produto desta dissertação também pode ser considerado uma contribuição para a Estação TABA, visto que o objetivo é prover um mecanismo eficiente e eficaz para avaliar propostas de melhoria de processos de software, tal como está prescrito em uma das atividades da Estratégia de Avaliação e Melhoria em Níveis adotada pela Estação TABA.

No capítulo seguinte será apresentada um estudo baseado em revisão sistemática, com o objetivo de identificar instrumentos que facilitem a realização de projetos-piloto, observando as práticas sugeridas pelos modelos de qualidade.

## **Capítulo 4. Instrumentos que Apóiam a Realização de Pilotos – Um Estudo Baseado em Revisão Sistemática**

---

### **4.1. Introdução**

A revisão bibliográfica apresentada no capítulo 2 apresentou modelos, normas e abordagens que sugerem um conjunto de práticas para introduzir mudanças em processos de software e a adoção de pilotos para avaliar a efetividade destas mudanças. Entretanto, os instrumentos identificados durante a revisão informal não contemplavam todas as práticas identificadas na literatura para a realização de pilotos.

Este fato motivou a realização de um estudo complementar, baseado em revisão sistemática, para a busca de outros instrumentos que utilizem pilotos para a avaliação da efetividade das mudanças introduzidas nos processos de software.

A revisão sistemática é um tipo de estudo secundário (KITCHENHAM, 2004), cujo processo de pesquisa segue um conjunto de passos metodologicamente bem definidos de acordo com um protocolo previamente desenvolvido (BIOLCHINI et al., 2005). Vale ressaltar que os procedimentos oriundos da revisão sistemática minimizam os riscos associados ao viés da revisão informal. Na revisão informal o pesquisador tende a utilizar como ponto de partida seu conhecimento sobre o assunto, ou seja, o desconhecimento sobre outras vertentes e perspectivas sobre a questão de pesquisa pode restringir e empobrecer os resultados obtidos.

Com a realização deste estudo esperam-se caracterizar o estado da arte relacionado com a utilização de pilotos para avaliar a efetividade das melhorias de processo de software e, quiçá, apresentar lacunas de conhecimento e/ou conhecimentos consolidados que motivam e servem como ponto de partida para novos estudos primários ou secundários.

Nas seções subseqüentes, será apresentada a estratégia utilizada para execução do estudo, o desenvolvimento do protocolo de pesquisa e dos instrumentos de apoio, além de aspectos inerentes à execução e resultados do estudo. Concluindo o capítulo, serão feitas considerações a respeito dos fatores que podem ter influenciado os resultados e das lições aprendidas.

### **4.2. Estratégia Utilizada**

A estratégia consistiu na elaboração de um protocolo preliminar simplificado e realização de refinamentos sucessivos para torná-lo adequado à pesquisa. As

atividades de refinamento incluíram a evolução da expressão de busca a ser aplicada, o exercício dos procedimentos de seleção dos estudos, a coleta de amostras e o aperfeiçoamento dos instrumentos a serem utilizados.

Os resultados do refinamento foram: o protocolo adequado à realização da pesquisa, instrumentos apropriados para registro e análise dos dados coletados.

### **4.3. Desenvolvimento do Protocolo de Pesquisa**

O primeiro momento do desenvolvimento do protocolo foi destinado a uma discussão que incluiu a motivação e a apresentação do problema a ser investigado. Após a clara definição do contexto, definiu-se a principal questão de pesquisa que orientou a realização do estudo:

Quais instrumentos têm sido propostos para facilitar a realização de pilotos no âmbito industrial com o objetivo de avaliar a efetividade das mudanças dos processos de software?

Além da questão principal, outras secundárias foram formuladas:

- Quais tipos de instrumentos<sup>10</sup> estão sendo propostos?
- Quais práticas propostas pelos modelos e normas em prol da qualidade são atendidas pelos instrumentos?
- Quais instrumentos tiveram seu uso avaliado na indústria?
- Quais instrumentos fornecem ou utilizam ferramentas de apoio?
- Qual o modelo ou norma em prol da qualidade mais referenciada por estes instrumentos?

A premissa para a realização da pesquisa foi que a existência das práticas sugeridas por modelos e normas em prol da qualidade (Tabela 4.1) pode potencializar os resultados obtidos na realização de pilotos.

Para responder a questão de pesquisa, optou-se pelo uso de bibliotecas digitais disponibilizadas por editoras especializadas em publicações científicas e definiram-se critérios para orientar esta seleção (vide Apêndice A, item 7).

A identificação das fontes de dados, isto é, das publicações utilizadas para extração dos dados, foi feita com base nas publicações contidas nas bibliotecas digitais selecionadas, constituídas por artigos técnicos apresentados em congressos e publicados em anais, relatos de experiência, artigos de periódicos, prospectos de teses, normas, dentre outras.

---

<sup>10</sup> Neste caso, instrumentos correspondem a procedimentos, processos, abordagens, técnicas, métodos, ferramentas ou quaisquer mecanismos que facilitem a realização de pilotos.

**Tabela 4.1 - Práticas para realização de pilotos sugeridas por modelos de qualidade.**

Práticas	Modelos/Normas	Descrição
Uso de indicadores	QIP, CMMI, IDEAL, ISO/IEC 15504, MR-MPS	Os indicadores podem ser quantitativos ou qualitativos. Quantitativos são obtidos a partir de métricas, que são selecionadas previamente para prover uma descrição quantitativa dos efeitos observados. Os indicadores qualitativos são oriundos de avaliações de especialistas, <i>post-mortem</i> , dentre outras;
Planejamento	CMMI, IDEAL, ISO/IEC 15504, MR-MPS	O piloto tem um propósito específico e, para atingir seus objetivos, é necessário alocar recursos e tomar todas as providências necessárias para que o piloto seja executado de forma permitir a observação dos efeitos das mudanças no processo.
Seleção da Equipe	IDEAL	Os participantes devem ser pessoas dispostas e capazes de aprender e executar as ações de melhoria propostas.
Treinamento da Equipe	CMMI, IDEAL, ISO/IEC 15504	A equipe deve ser treinada para que as ações propostas sejam executadas conforme descritas.
Uso de mais de um projeto	CMMI, IDEAL, ISO/IEC 15504, MR-MPS	O número de projetos pode aumentar a visibilidade dos efeitos e dos fatores que exerceram influência nos resultados, contribuindo com o refinamento da proposta de melhoria.
Registro de Comprometimento	CMMI	Obter o comprometimento das pessoas é um dos fatores de sucesso em qualquer projeto. No caso dos pilotos, requer uma atenção especial, pois em geral exige mudanças na execução das atividades.
Adequação do Ambiente de Operação	CMMI, ISO/IEC 15504	Um piloto executado em um ambiente diferente daquele que os projetos são realizados pode não estar sujeito a todas as intempéries do cotidiano da organização e, conseqüentemente, a avaliação poderá retratar um resultado distorcido.
Execução monitorada	CMMI, IDEAL, ISO/IEC 15504	É preciso verificar se durante a operação o ambiente está conforme o planejado e se as medidas estão sendo coletadas adequadamente.
Avaliação dos resultados	QIP, CMMI, IDEAL, ISO/IEC 15504, MR-MPS	Análise quantitativa e/ou qualitativa dos resultados obtidos com o piloto.
Documentação dos Resultados	QIP, CMMI, IDEAL, ISO/IEC 15504, MR-MPS	Ao final, todo conhecimento explicitado (problemas encontrados e lições aprendidas) pelo piloto deve ser devidamente registrado e disponibilizado para a organização.

Vale ressaltar que não foram estabelecidos procedimentos para validação dos instrumentos apresentados nas fontes, pois o interesse desta pesquisa restringia-se apenas à identificação e caracterização destes e não na efetividade dos resultados obtidos a partir do uso destes instrumentos.

A definição da expressão de busca foi um dos itens que exigiu maior esforço para elaboração do protocolo. A primeira preocupação foi na identificação de termos que fossem precisos o suficiente para permitir a seleção de todas as publicações relacionadas à questão de pesquisa e que, ao mesmo tempo, restringissem os resultados da seleção apenas às publicações de interesse. Da versão preliminar para a versão final do protocolo houve uma evolução significativa da expressão.

A primeira versão da expressão de busca foi obtida a partir de termos chaves extraídos da definição da população do estudo, da intervenção a ser aplicada e dos resultados esperados (PAI et al., 1995). Os termos foram combinados com sinônimos e palavras correlatas e deram origem a seguinte expressão:

*(pilot or trial or test) and  
(evaluation or evaluating or assessment or appraisal or testing) and  
(change or changing or alter) and  
“software process” and  
(improvement or improving or evolution) and  
(process or methodology or technique or approach or tool)*

Esta expressão foi refinada e avaliada sucessivas vezes por meio da realização de ciclos de teste. Cada ciclo de teste consistiu na aplicação do protocolo e análise de uma pequena amostra. A expressão de busca resultante foi a seguinte:

*“software process improvement” and pilot*

Algumas decisões tiveram que ser tomadas durante o refinamento da expressão de busca. Dentre elas, a opção pela construção de uma expressão pouco restritiva, ou seja, que implicasse na diminuição do risco de exclusão de publicações relevantes. Esta opção teve como consequência direta o aumento significativo do número de publicações, o que implicou o incremento do esforço na etapa de execução.

A cada ciclo de refinamento da expressão de busca, os procedimentos de coleta e os instrumentos de apoio também foram evoluídos para que a execução do estudo e a análise fossem facilitadas. Um banco de dados foi projetado e construído para permitir o registro de bibliotecas e fontes, apoiar as etapas de seleção das publicações e armazenar de forma estruturada os dados extraídos dos documentos da revisão.

O desenvolvimento do protocolo teve início em 17 de novembro de 2005 e se estendeu até o último dia do ano. As atividades contabilizaram aproximadamente 60 horas de trabalho.

#### **4.4. Desenvolvimento dos Instrumentos de Apoio**

Os instrumentos de apoio foram desenvolvidos em paralelo a execução dos ciclos de teste. À medida que cada ciclo era executado, os instrumentos eram melhorados para facilitar o registro dos dados coletados e, posteriormente, facilitar a análise dos achados.

O primeiro instrumento a ser construído foi um banco de dados para armazenar, de forma relacionada, publicações, fontes e bibliotecas envolvidas no estudo. Com a evolução do protocolo foram incluídos alguns atributos e outras entidades que não haviam sido previstas *a priori*. A Figura 4.1 ilustra o modelo de entidade-relacionamento do banco de dados construído.

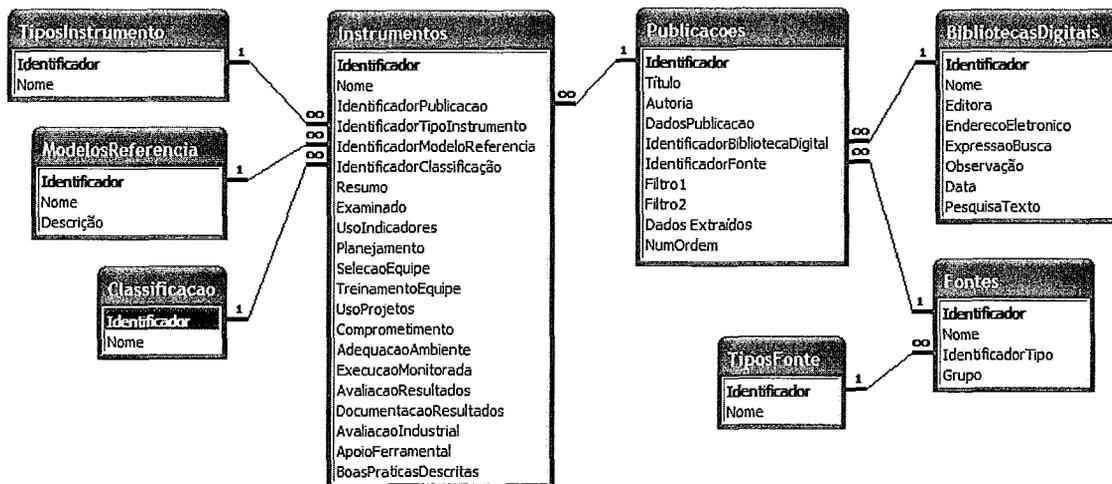


Figura 4.1 - Modelo entidade-relacionamento do banco de dados de apoio à revisão sistemática.

Além do banco de dados, formulários para coleta de dados e relatórios também foram elaborados. A Figura 4.2 ilustra o formulário utilizado para auxiliar a etapa de extração dos dados das publicações selecionadas.

Figura 4.2 - Formulário para apoiar a extração dos dados.

## 4.5. Execução do Estudo

A execução do estudo foi dividida em três etapas. A primeira etapa faz uso dos engenhos de busca para selecionar publicações que podem estar relacionadas com a questão de pesquisa. Esta busca é meramente sintática. As duas etapas subsequentes tratam de selecionar semanticamente as publicações, verificando os

documentos que realmente estão relacionados com a questão de pesquisa e analisando aqueles que possuem dados relevantes a serem extraídos (vide Apêndice A, item 11).

O ponto de partida para execução do estudo foi a seleção das bibliotecas digitais e seus respectivos engenhos de busca. Com a aplicação dos critérios definidos no protocolo (Apêndice A, item 7), obteve-se a lista de bibliotecas ilustrada na Tabela 4.2.

**Tabela 4.2 - Bibliotecas digitais selecionadas.**

Biblioteca	Organização	Endereço eletrônico
ACM Digital Library	ACM	<a href="http://portal.acm.org">http://portal.acm.org</a>
BlackWell Sinergy	BlackWell	<a href="http://www.blackwell-synergy.com/">http://www.blackwell-synergy.com/</a>
Cambridge Journals Online	Cambridge Journals	<a href="http://journals.cambridge.org">http://journals.cambridge.org</a>
Emerald Insight	Emerald	<a href="http://www.emeraldinsight.com">http://www.emeraldinsight.com</a>
HighWire	HighWire Press (Stanford University)	<a href="http://www.highwire.org">http://www.highwire.org</a>
IEEE Xplore	IEEE	<a href="http://ieeexplore.ieee.org">http://ieeexplore.ieee.org</a>
Oxford Journals	Oxford University	<a href="http://www.oxfordjournals.org/">http://www.oxfordjournals.org/</a>
ProQuest	ProQuest	<a href="http://proquest.umi.com">http://proquest.umi.com</a>
Sage Journals Online	Sage	<a href="http://online.sagepub.com/">http://online.sagepub.com/</a>
Science Direct Online	Elsevier Science Inc.	<a href="http://www.sciencedirect.com">http://www.sciencedirect.com</a>
SpringerLink	Springer	<a href="http://www.springerlink.com">http://www.springerlink.com</a>

Outras bibliotecas foram identificadas (Web of Science, Compendex e CSA Illumina), porém, os respectivos engenhos de busca não permitiam a pesquisa no texto completo das publicações. Este foi um dos critérios para seleção das bibliotecas.

Dentre as excluídas, a Compendex se destaca por indexar um número expressivo de publicações em engenharia. Contudo, ao realizar uma busca a título de experiência, o resultado foi um conjunto de cinco publicações, das quais três não possuíam nenhum dado relevante para a questão de pesquisa, uma estava acessível apenas através da aquisição (editora Wiley InterScience) e a outra não estava disponível na editora (Elsevier; faltava exatamente a publicação do volume).

As bibliotecas digitais apresentaram peculiaridades nas opções de busca que tiveram de ser tratadas individualmente. Por exemplo, para utilizar o engenho de busca da ACM Digital Library a expressão teve que ser alterada.

A Tabela 4.3 apresenta as bibliotecas selecionadas e as respectivas expressões de busca e opções aplicadas.

**Tabela 4.3 - Expressões de busca e opções aplicadas nas bibliotecas digitais.**

Biblioteca	Expressão de busca	Opções
ACM Digital Library	+"software process improvement" +pilot	Busca avançada. Opção "All Information", que inclui pesquisa no texto completo. A opção "Results must have accessible: Full Text" também foi selecionada.
BlackWell Sinergy	"software process improvement" AND pilot	Busca avançada. Opção "Full Text"
Cambridge Journals Online	"software process improvement" AND pilot	Busca avançada. Opções: "All journals"; Subject: "All categories".

Biblioteca	Expressão de busca	Opções
Emerald Insight	"software process improvement" AND pilot	Busca avançada. Opções: "All fields" (inclui a pesquisa por texto completo)
HighWire	"software process improvement" AND pilot	Opção: "Anywhere in Text"
IEEE Xplore	"software process improvement" AND pilot	Busca avançada. Opção: "Full Text & All Fields"
Oxford Journals	"software process improvement" AND pilot	Busca avançada; Opções: "Full Text   Abstract   Title"; Subject area: "All"
ProQuest	"software process improvement" AND pilot	Busca avançada. Opção: Documento completo;
Sage Journals Online	"software process improvement" AND pilot	Opção: "Full Text   Abstract   Title"
Science Direct Online	"software process improvement" AND pilot	Opções de busca: "Full Text"; in "Journals, Book Series, Handbooks, Abstract"; Database; "All years";
SpringerLink	"software process improvement" AND pilot	Busca avançada de artigos por texto. Opções: "Boolean Search"; "Full Text";

Através da consulta ao manual de uso da biblioteca digital IEEE Xplore foi identificada uma forma de aplicar a expressão de busca sem que fosse necessário optar pela busca avançada. Porém, foi obtido um número inferior de publicações.

Durante a catalogação preliminar foram identificadas 95 fontes distintas (vide Apêndice B.). As dez fontes que mais contribuíram com publicações foram: Computer (10), Journal of Systems and Software (11), Empirical Software Engineering (12), IEEE Transactions on Software Engineering (12), International Symposium Empirical Software Engineering (14), Software Quality Journal (15), Lecture Notes in Computer Science (20), Annual Hawaii International Conference on System Sciences (24), International Conference on Software Engineering (31) e IEEE Software (34).

A primeira etapa da seleção resultou em 399 documentos, coletados a partir das bibliotecas selecionadas, conforme ilustrado na Tabela 4.4. Desse total, 58 publicações não estavam disponíveis através das bibliotecas digitais, pois o acesso estava condicionado ao pagamento de tarifas. A grande maioria destas publicações consta na série Lectures Notes in Computer Science (Spring-Verlag).

**Tabela 4.4 – Total de documentos por biblioteca – seleção e catalogação preliminar**

Biblioteca	Número de Publicações		
	Disponível	Indisponível	Total
ACM Digital Library	53	0	53
BlackWell Sinergy	1	0	1
Cambridge Journals Online	0	0	0
Emerald Insight	7	0	7
HighWire	0	0	0
IEEE Xplore	174	0	174
Oxford Journals	0	0	0
ProQuest	43	0	43
Sage Journals Online	1	0	1
Science Direct Online	19	3	22
SpringerLink	43	55	98
Totais	341	58	399

Observando o escopo da pesquisa, que restringe a análise às publicações obtidas a partir das próprias bibliotecas digitais e que o acesso a estas não deve implicar ônus para o estudo, 58 publicações indisponíveis nas bibliotecas digitais não foram incluídas (Apêndice E). Esta exclusão reduziu o número de publicações para 341.

A atividade de seleção e catalogação preliminar das publicações teve início no primeiro dia do ano de 2006 e consumiu aproximadamente 36 horas.

Antes de iniciar a segunda etapa, foi realizado um trabalho com o objetivo de identificar publicações duplicadas, isto é, aquelas obtidas a partir de mais de uma biblioteca (vide lista no Apêndice C). Através de consultas ao banco de dados, 25 duplicações foram identificadas. Assim, o número de publicações a serem examinadas passou de 341 para 316 (Apêndice D). Vale ressaltar que uma destas duplicações só foi detectada durante a análise, pois o título listado na biblioteca ACM estava errado. Ao invés de “From MCC to CMM: technology transfers bright and dim” a publicação estava listada como “From MCC and CMM: technology transfers bright and dim”.

Na última semana de janeiro, foi iniciada a análise das publicações com relação aos critérios estabelecidos pelo primeiro filtro. As publicações deveriam tratar de alguma alteração realizada em processos de software em prol da melhoria e utilizar ou recomendar o uso de projetos-piloto para avaliar a efetividade dessas alterações. Nesta etapa, 35 publicações (Apêndice F), oriundas de 23 fontes (Tabela 4.5), foram selecionadas.

**Tabela 4.5 – Total de publicações selecionadas na segunda etapa por fonte.**

Fontes	Total
ACM SIGAda Ada Letters	1
ACM SIGSOFT Software Engineering Notes	1
Annual Hawaii International Conference on System Sciences	3
Annual International Computer Software and Applications Conference	1
Centre for Advanced Studies on Collaborative research: software engineering	1
Computer	1
Empirical Software Engineering	1
Euromicro Conference	1
IEEE Joint International Conference on Requirements Engineering	1
IEEE Nuclear Science Symposium Conference Record	1
IEEE Software	2
IEEE Std 1348-1995	1
IEEE Transactions on Software Engineering	1
Information and Software Technology	2
Information Technology & People	1
International Conference on Software Engineering	6
International Symposium on Software Metrics	1
Journal of Systems and Software	2
Lecture Notes in Computer Science	3
Mexican International Conference on Computer Science	1
Portland Int Conference on Mgmt and Tech. Innovation in Tech. Management (PICMET)	1
Software Engineering Conference	1
Software Quality Journal	1
<b>Total</b>	<b>35</b>

Na terceira etapa, as 35 publicações selecionadas foram analisadas em busca da identificação de instrumentos que apoiassem a realização de pilotos em ambiente industrial, conforme critério estabelecido no protocolo (vide Apêndice A, item 11, subitem iii). Apenas 5 atenderam aos critérios, vide Tabela 4.6.

**Tabela 4.6 – Publicações selecionadas para extração de dados**

Título	Autoria	Fonte
Designing and Conducting na Empirical Study on Test Management Automation	Griselda Giraudo, Paolo Tonella	Empirical Software Engineering, Volume 8, Issue 1, Mar 2003, pp.59-81
Experiences from the pilot operation and commissioning phase of a SCM process improvement program	Nattinen, M.; Rahikkala, T.; Valimaki, A.,	EUROMICRO Conference, 1999. Proceedings. 25th , vol.2, no.pp.185-192 vol.2, 1999
IEEE Std 1348-1995. IEEE recommended practice for the adoption of Computer-Aided Software Engineering (CASE) tools	IEEE	IEEE Std 1348-1995
Implementing requirements engineering processes throughout organizations: success factors and challenges	Marjo Kauppinen, Matti Vartiainen, Jyrki Kontio, Sari Kujala and Reijo Sulonen	Information and Software Technology, Volume 46, Issue 14, 1 November 2004, Pages 937-953
TIM: a tool insertion method	Tilmann Bruckhaus	Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Canada, Page: 7, 1994

As publicações selecionadas para a extração dos dados foram classificadas, conforme prescrito no protocolo (Apêndice A, item 12.3), resultando no conjunto de registros ilustrados na Tabela 4.7 .

**Tabela 4.7 – Registro preliminar das publicações com dados para coleta**

Referência	Classificação	Tipos de Instrumento	Modelo de Referência	Resumo
Giraudo e Tonella (2003)	Relato de Experiência	Conjunto de Procedimentos	Método Científico	Inclusão de nova ferramenta para automação de testes em um processo de software existente na organização. Para avaliar os efeitos, foram utilizados três projetos-piloto em um estudo experimental baseado nas propostas de Basili e Kitchenham.
Nattinen et al. (1999)	Relato de Experiência	Abordagem	PRIMER <sup>11</sup>	O artigo trata de uma abordagem sistemática para análise e melhoria do processo de gerência de configuração. A abordagem consiste em 4 fases: caracterizar a situação, estabelecer objetivos, planejar as ações de melhoria e pilotar as melhorias.
IEEE (1995)	Proposta	Processo	IEEE Std 1209-1992	A IEEE prescreve um processo para facilitar adoção de tecnologias CASE para prover melhorias no processo de desenvolvimento e manutenção. Uma das macro-atividades consiste na condução de um Piloto, que sugere os passos: determinar as características do piloto, planejar o piloto, executar o piloto, avaliar o piloto e tomar a decisão sobre a adoção da ferramenta.

<sup>11</sup> PR<sup>2</sup>IMER - Practical Process Improvement for Embedded Real-time Software, é um arcabouço para melhoria de processo de software desenvolvida pela empresa VTT Electronics.

Referência	Classificação	Tipos de Instrumento	Modelo de Referência	Resumo
Kauppinen et al. (2004)	Relato de Experiência	Conjunto de Procedimentos	ISO/IEC 15504	Relato de experiência em prol da melhoria do processo de engenharia de requisitos. Em uma das etapas, a utilização de projetos pilotos é proposta para avaliar as melhorias incluídas no processo. Os benefícios alcançados com a utilização dos pilotos no ambiente industrial são descritos em forma de lições aprendidas, porém as práticas não são descritas. A publicação apenas indica as fontes das práticas para a realização de pilotos.
Bruckhaus (1994)	Proposta	Método	QIP	O método propõe o uso de pilotos para facilitar a introdução de novas ferramentas nos processos de software e um programa de medição que permite acompanhar o progresso da inserção, aferir o alcance dos objetivos e avaliar os benefícios obtidos.

Por fim, os dados foram extraídos das cinco publicações selecionadas. A extração, conforme declarada no protocolo (Apêndice A, item 12.4), consistiu em verificar quais práticas sugeridas por modelos e normas em prol da qualidade que o instrumento atende, se o instrumento teve seu uso avaliado na indústria e se o instrumento fornece ou utiliza ferramentas de apoio. O sumário da extração dos dados está ilustrado na Tabela 4.8.

**Tabela 4.8 - Publicações selecionadas e características dos respectivos instrumentos de apoio à realização de pilotos**

Título	Uso de Indicadores	Planejamento	Seleção da Equipe	Treinamento da Equipe	Piloto em mais de um Projeto	Comprometimento	Preparação do Ambiente	Execução Monitorada	Avaliação dos Resultados	Documentação dos Resultados	Avaliação na Indústria	Apoio de Ferramentas	Práticas Descritas em detalhes
Designing and Conducting an Empirical Study on Test Management Automation	✓	✓	✓	-	✓	-	✓	-	✓	✓	✓	-	✓
Experiences from the pilot operation and commissioning phase of a SCM process improvement program	✓	✓	-	-	✓	-	-	-	✓	✓	✓	-	-
IEEE Std 1348-1995. IEEE recommended practice for the adoption of Computer-Aided Software Engineering (CASE) tools	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓
Implementing requirements engineering processes throughout organizations: success factors and challenges	-	-	-	-	✓	-	-	-	-	-	✓	-	-
TIM: a tool insertion method	✓	-	-	✓	-	-	✓	-	✓	✓	-	-	-

A execução do estudo consumiu aproximadamente 160 horas, incluindo as atividades de seleção, análise e extração dos dados.

## 4.6. Análise dos Dados Coletados

O percentual das publicações que continham dados para a coleta foi de apenas 1,57% do total selecionado na primeira etapa (316 publicações) ou 14,28% do total selecionado na segunda etapa (35 publicações).

Tabela 4.9 - Totalização de publicações, fontes e bibliotecas por etapa

	Pré-seleção	1ª Etapa	2ª Etapa	3ª Etapa
Total de Publicações	399	316	35	5
Total de Fontes	125	95	23	5
Total de Bibliotecas Digitais	11	8	5	4

No contexto deste estudo, apenas cinco instrumentos que facilitam a realização de pilotos no âmbito industrial e que tem como objetivo avaliar a efetividade das mudanças dos processos de software foram encontrados: dois conjuntos de procedimentos, um processo, uma abordagem e um método.

Apenas um dos instrumentos identificados atendeu a todas as práticas utilizadas para caracterização, o contido em IEEE (1995). Porém, o instrumento é restrito à condução de pilotos para introdução de tecnologias CASE nas organizações de desenvolvimento e manutenção de software. O instrumento que menos atendeu às práticas, contido em Kauppinen et al. (2004), fez menção a outras publicações que, supostamente, serviram de guia para a condução das atividades relatadas.

Três instrumentos relataram sua aplicação no âmbito industrial, nas áreas: automação de testes (GIRAUDO; TONELLA, 2003), engenharia de requisitos (KAUPPINEN et al., 2004) e gerência de configuração (NATTINEN et al., 1999). O IEEE (1995) foi apresentado na forma de padrão internacional da IEEE, portanto, a publicação não apresentou avaliação industrial. O instrumento na forma de método, apresentado por Bruckhaus (1994), ainda está na forma de proposta e também não apresentou avaliação na indústria.

Nenhum dos instrumentos apresentou ferramentas para apoiar a realização de pilotos.

O processo descrito em IEEE (1995) e o conjunto de procedimentos apresentados por Giraud e Tonella (2003) forneceram um bom nível de detalhes sobre as práticas. Diferente dos demais instrumentos, que apresentaram apenas descrições sucintas, o IEEE (1995) chega a propor alguns modelos de artefatos que podem ser utilizados para apoiar a execução das atividades.

Embora o Quality Improvement Paradigm tenha sido relacionado apenas uma vez como modelo de referência neste pequeno conjunto de publicações, pode-se afirmar que o mesmo é o mais presente. Pois, o modelo PR<sup>2</sup>IMER referenciado por Nattinen et al. (1999) também foi inspirado no QIP.

A atividade de análise consumiu 5 horas e consistiu na verificação dos dados extraídos, análise dos dados e no relato dos resultados (elaboração deste item).

## **4.7. Considerações Finais**

### **4.7.1. Fatores que podem ter influenciado a pesquisa**

Uma das dificuldades encontradas durante a realização do estudo foi a desconcentração provocada por boas publicações. Durante a análise, foi comum encontrar textos interessantes em áreas correlatas que, por vezes, provocou um atraso na leitura dos demais trabalhos.

A exclusão de 58 artigos do estudo, por restrição de acesso nas bibliotecas digitais, pode ter exercido influência no resultado obtido no estudo. Principalmente pelo fato da maioria das publicações terem sido publicadas em eventos como o PROFES<sup>12</sup>, uma referência importante no que tange a publicações em melhoria de processos de software.

A identificação dos instrumentos facilitadores para a realização de pilotos foi realizada a partir da análise de experiências e propostas, cujo foco não estava centrado na utilização de pilotos para avaliar a efetividade das melhorias de processo de software. Este fato dificultou a identificação dos instrumentos, visto que, geralmente, estes não estavam explícitos.

As 35 publicações selecionadas na segunda etapa, em geral, apresentavam programas ou projetos de melhoria e quais os benefícios obtidos a partir da sua utilização. É provável que o relato pormenorizado do uso de pilotos, neste contexto, fique em segundo plano. Assim, a restrição de espaço para publicação pode ter levado a supressão dos instrumentos utilizados para condução de pilotos em uma das etapas da iniciativa.

### **4.7.2. Questões em aberto e algumas hipóteses**

O resultado obtido propõe algumas reflexões:

As organizações realizam pilotos observando as práticas sugeridas por modelos em prol da qualidade? Será que há conhecimento suficiente nas organizações para permitir a realização de pilotos de forma acurada? As organizações estão cientes dos benefícios na utilização adequada do uso de pilotos?

Embora os modelos proponham práticas para realização de pilotos, os mesmos não descrevem em detalhes as atividades que devem ser executadas. Portanto, o fato

---

<sup>12</sup> International Conference on Product Focused Software Process Improvement

das organizações atenderem as práticas sugeridas pelos modelos não significa que os resultados obtidos a partir dos pilotos representem de forma acurada os efeitos das mudanças realizadas nos processos de software.

Mesmo que as organizações tenham ciência a respeito dos benefícios da realização de pilotos, aspectos relacionados ao consumo de esforço e custo para planejamento, execução e análise dos pilotos podem desestimular estas iniciativas.

#### **4.7.3. Lições aprendidas**

É imprescindível avaliar o funcionamento dos mecanismos de busca para identificar quais as opções padrão e que outras estão disponíveis para facilitar a busca e, ao mesmo tempo, evitar retrabalho.

O desenvolvimento do protocolo provê conhecimento sobre o domínio associado à questão de pesquisa e temas correlatos. Através da aplicação de ciclos de teste do protocolo, é possível observar relações do tema de interesse com tópicos periféricos. Estas relações podem apoiar o refinamento do protocolo e impactar no redimensionamento do esforço para a execução da pesquisa.

Durante a execução do estudo é importante assumir uma perspectiva centrada na questão de pesquisa e nos critérios de seleção prescritos no protocolo. Momentos de desconcentração podem implicar no desperdício de esforço e tempo, além do custo da troca de contexto para a retomada dos trabalhos.

É recomendável destinar um intervalo contínuo do tempo para a realização deste tipo de estudo. Interrupções podem ser prejudiciais ao desempenho. A partir de um determinado momento, a questão de pesquisa e os critérios ficam bem sedimentados na memória do revisor. Isto diminui a necessidade de recorrer ao protocolo para certificar-se dos critérios. Portanto, manter as questões e critérios em mente pode auxiliar na diminuição do esforço e do tempo destinado a realização do estudo.

O esforço requerido para realizar um estudo baseado em revisão sistemática não deve ser subestimado.

#### **4.7.4. Conclusões**

O esforço total para realização do estudo foi de aproximadamente 225 horas, sendo 60h para o desenvolvimento do protocolo e instrumentos de apoio, 160h para a execução do estudo e 5h para a análise dos dados.

Há indícios de que a utilização de pilotos para avaliar a efetividade das propostas de melhoria produz bons resultados. 35 publicações, das 316 analisadas sugerem e/ou utilizam pilotos com este propósito.

Nenhum dos instrumentos identificados no estudo apresentou ferramentas de apoio à realização de pilotos. O instrumento que mais atendeu às práticas de caracterização utilizadas neste estudo possui caráter específico, isto é, é voltado à introdução de tecnologias CASE nas organizações.

Os instrumentos identificados neste estudo não haviam sido capturados durante a realização da revisão informal da literatura, apresentada no capítulo 2 desta dissertação.

O capítulo seguinte propõe uma abordagem que visa orientar as organizações na realização pilotos para avaliar propostas de melhoria de processos de software, antes que as mudanças nos processos sejam institucionalizadas. A abordagem atende às práticas sugeridas por modelos de qualidade de software no que tange a realização de pilotos e também provê mecanismos facilitadores para potencializar os resultados das experiências com projetos-piloto.

# **Capítulo 5. Uma Abordagem Baseada em Experimentação para Avaliação da Efetividade de Propostas de Melhoria de Processos de Software**

---

## **5.1. Introdução**

A utilização de pilotos é uma prática observada tanto no ambiente científico como no ambiente industrial. No ambiente científico, geralmente, o piloto precede a execução de estudos bem elaborados, enquanto no ambiente industrial ele precede a implantação de mudanças com relativo impacto, riscos e incertezas associadas.

No capítulo 3 foi apresentada uma revisão sistemática sobre o uso de pilotos na melhoria de processos de software que identificou um número reduzido de instrumentos que apóiam a realização de pilotos em ambiente industrial.

A compilação dos dados coletados no abrangente levantamento bibliográfico combinada à experiência obtida a partir de estudos realizados na indústria, deu origem a uma abordagem, baseada em experimentação, para apoiar a realização de pilotos em prol da avaliação da efetividade das propostas de melhoria de processos de software.

Nas próximas seções deste capítulo será apresentada a abordagem e seus elementos constituintes.

## **5.2. A Abordagem Baseada em Experimentação**

Avaliar a efetividade das propostas de melhoria através do uso de projetos-piloto não é tarefa trivial. À primeira vista, a realização de um piloto consiste em aplicar a proposta de melhoria em um projeto e verificar seus resultados. Porém, há um conjunto de considerações que precisam ser feitas para se executar um piloto e obter acurácia na avaliação da proposta. Dentre elas:

O projeto selecionado é representativo com relação aos demais projetos da organização que serão contemplados com as ações que estão sendo testadas?

Um projeto é suficiente para avaliar a proposta?

As pessoas precisam ser treinadas? Em que?

Qual aspecto do projeto que está associado à proposta a ser testada? Como estes aspectos podem ser mensurados?

Como os resultados do projeto podem conduzir à avaliação da efetividade da proposta de melhoria? O que se espera do projeto para afirmar se a proposta é aplicável ou não à organização?

Que fatores podem interferir na condução do projeto e afetar a avaliação da proposta de melhoria? O que pode ser feito para minimizar estas influências?

A proposta de melhoria pode ser refinada?

Com o objetivo de tratar estas questões, foi definida uma abordagem, baseada em experimentação, para realização de pilotos em prol da avaliação da efetividade das propostas de melhorias no âmbito industrial.

Conduzir um projeto-piloto capaz de testar as ações de melhoria sem interferir na logística e nos aspectos relacionadas a custo, disponibilidade de pessoal e infraestrutura é um desafio, pois a inclusão de novas atividades em um projeto tem impacto direto no cronograma e na alocação de recursos. A abordagem proposta foi concebida para minimizar consideravelmente estes fatores e, ao mesmo tempo, potencializar os resultados obtidos na avaliação da efetividade das melhorias.

A abordagem descrita nesta dissertação é dotada de características que a diferenciam das demais experiências identificadas durante a revisão sistemática no capítulo 3. Dentre as mais significativas estão: (i) atendimento às práticas sugeridas por modelos de qualidade; (ii) existência do apoio ferramental; e (iii) possibilidade de interagir com ambientes de engenharia de software centrados em processos.

Três elementos compõem a abordagem: (1) Um processo, que orienta as ações a serem executadas durante a realização do piloto; (2) Uma ferramenta, que apóia a execução das atividades do processo; e (3) Um guia, que contém conhecimento sobre o processo e sobre a ferramenta e serve de roteiro para permitir a utilização eficiente da abordagem.

### **5.3. O Processo**

O processo foi o primeiro elemento a ser definido. Após uma experiência na indústria com o intuito de avaliar a viabilidade de sistematizar a realização de pilotos nas organizações (SILVA FILHO et al., 2005), os esforços da pesquisa se intensificaram no levantamento bibliográfico da literatura para identificar processos, procedimentos, atividades e outros instrumentos que apoiassem a realização de pilotos. Os achados constam descritos nos capítulos 2 e 3.

### 5.3.1. Elicitação das atividades do processo

Com base nos achados na literatura e na experiência vivenciada na condução de pilotos, foi dada seqüência à definição de um processo que combinasse as boas práticas providas pelos modelos de qualidade para a realização de pilotos à eficácia da execução de um processo experimental.

A Tabela 5.1 apresenta o conjunto inicial de atividades oriundas do processo convencional de experimentação em engenharia de software, concebido com base nos trabalhos Kitchenham et al. (1995), Wohlin et al. (2000), Kitchenham et al. (2002), Travassos et. al (2002) e Jedlitschka (2005).

**Tabela 5.1 – Conjunto de atividades elicítadas.**

	Atividades	Descrição
Definição	Descrever o Problema	Identificar claramente a proposta de melhoria e quais os problemas que estão sendo tratados.
	Determinar os Objetivos	Determinar os objetivos para o teste da melhoria, que deve ser avaliar a efetividade da proposta com relação a algum aspecto específico.
Planejamento	Selecionar Contexto	Caracterizar o contexto ao qual a proposta se aplica e as ações de melhoria que serão testadas.
	Formular Hipóteses	Declarar a hipótese nula e as hipóteses alternativas a serem verificadas ao final do estudo.
	Selecionar Variáveis	Identificar as variáveis dependentes e as independentes, que deverão ser mensuradas e monitoradas ao longo da execução do estudo, bem como declarar a escala e procedimentos de coleta que serão aplicados a cada uma delas.
	Escolher o <i>Design</i>	Verificar quantos tratamentos serão aplicados às variáveis dependentes do estudo.
	Selecionar Participantes	Caracterizar o perfil e selecionar os participantes mais adequados.
	Determinar Instrumentação	Enumerar todos os instrumentos necessários à condução do estudo, desde a tecnologia que irá ser testada até as ferramentas que apoiaram a coleta e análise dos dados.
	Tratar a Validade	Identificar questões que poderão restringir a validade do estudo e tratá-las adequadamente para que os resultados do estudo tenham significado amplo considerando o contexto ao qual a solução proposta se aplica. As questões de validade também envolvem a identificação dos fatores de confusão que podem dificultar a visualização dos efeitos proporcionados pelas ações que estão sendo testadas.
Operação	Preparar para Execução	Treinar, motivar e comprometer os participantes, preparar o ambiente para operação e coleta dos dados.
	Executar	Iniciar a execução da avaliação através da monitoração constante das variáveis e do ambiente, obtendo feedback constante dos participantes e observando os fatores que estão influenciando as ações sendo testadas.
	Mensurar	Coletar medidas das métricas para cômputo das variáveis dependentes e observação das variáveis independentes.
	Validar Dados	Aplicar procedimentos para validar os dados que estão sendo coletados para garantir sua acurácia.
Análise e Interpretação	Adequar Dados à Análise	Tornar os dados passíveis de comparação com séries históricas (redução e normalização) e adequá-los à verificação das hipóteses.
	Teste de Hipóteses	Verificar as hipóteses declaradas no plano do estudo aplicando testes estatísticos.
	Emitir Conclusões	Realizar uma avaliação considerando o teste de hipóteses e os dados qualitativos coletados durante a operação do estudo.

Apresentação e Empacotamento	Formatar Resultados	Tabular e formatar os resultados para que eles possam ser compreensíveis aos participantes e interessados.
	Emitir Relatório Final	Relatar o resultado da avaliação da proposta de melhoria.
	Registrar Lições Aprendidas	Realizar o registro na base de lições aprendidas e das relações causa-efeito identificadas a partir do teste da proposta de melhoria.

### 5.3.2. Adaptação das atividades elicitadas às necessidades da indústria

As adaptações foram realizadas com o intuito de adequar as atividades do processo experimental ao uso na indústria de software para a realização de pilotos. O objetivo das adaptações é suplantar restrições observadas nas organizações quanto à disponibilidade de conhecimento relacionado à experimentação, quanto à capacidade de coletar dados e estabelecer bases históricas passíveis de comparação e devido à inexperiência na identificação de fatores que poderiam influenciar os resultados da avaliação.

As adaptações foram realizadas nas atividades consideradas mais complexas, que exigiriam do responsável pela execução do processo conhecimento específico sobre experimentação, estatística e experiência. Também foram consideradas as dificuldades operacionais, em termos de esforço e logística. Parte destas questões também foi tratada pelos demais elementos da abordagem.

A adaptação do conjunto inicial de atividades implicou a inclusão, alteração, exclusão, fusão e divisão de atividades. As razões para adequação de cada atividade são apresentadas a seguir:

#### A) Descrever o problema + Determinar os objetivos => Identificar Piloto

Considerando que uma das entradas do processo é a proposta de melhoria resultante de um trabalho prévio na identificação de oportunidade de melhoria e priorização das ações a serem realizadas, pressupõe-se que a proposta deve apresentar o problema claramente e os objetivos a serem alcançados com sua execução. Portanto, estas duas atividades iniciais foram concatenadas e simplificadas, de forma que o executor apenas extraia o propósito e as expectativas gerais da proposta. A existência destas informações é um dos critérios de entrada para início do processo.

#### B) Selecionar Contexto + Seleção dos Participantes => Estabelecer Critérios para Seleção do Projeto

A descrição do contexto ao qual as ações de melhoria se aplicam também faz parte da proposta de melhoria. Ela deve conter informações que permitam identificar

para que tipo de projeto, ambiente e equipe as ações de melhoria devem ser aplicadas.

A seleção dos participantes, *a priori*, deveria ser aleatória e representativa com relação à população. Entretanto, dependendo da proposta de melhoria, pode ser necessária uma seleção por conveniência. Isto é, determinadas ações a serem realizadas durante o teste da melhoria podem ser aplicáveis apenas a gerentes inexperientes, equipes com muita experiência em determinada tecnologia ou com conhecimento mediano em engenharia de software, etc.

As atividades de seleção do contexto e dos participantes estão associadas à caracterização do contexto geral ao qual o piloto será aplicado, portanto estas foram fundidas e deram origem à atividade Estabelecer Critérios para Seleção.

#### C) Formular Hipóteses => Registrar Expectativas da Avaliação

A formulação de hipóteses requer conhecimento sobre experimentação. Conceitos como hipótese nula, hipóteses alternativas, refutação e outros, são necessários para executar a atividade. Além de não ser trivial declarar hipóteses de forma consistente e passível de verificação, a representação formal pode requerer habilidades adicionais. Com a alteração da atividade, ao invés de formular hipóteses, o executor terá que declarar, quantitativamente, quais são as expectativas para os projetos que irão testar o piloto com relação à melhoria proposta.

Com a transformação da atividade, optou-se também pelo uso de um nome mais inteligível e na reordenação da atividade no novo processo, pois a declaração da expectativa deve ser realizada após a identificação das variáveis que se pretende controlar, conforme descrito a seguir.

#### D) Selecionar Variáveis => Elaborar Plano de Medição

A seleção de variáveis no processo original consistia na identificação das variáveis independentes e dependentes do estudo. As variáveis dependentes ou variáveis de resposta são aquelas que apresentam os efeitos aos tratamentos realizados. As independentes são aquelas que são controladas no decorrer do estudo.

No contexto da realização do piloto, o objeto de estudo é o processo. Os tratamentos consistem nas alterações no processo, aplicadas nas atividades que o compõem. Os efeitos da mudança nos processos serão observados através da mensuração das métricas durante a execução dos projetos. Estas métricas, associadas diretamente às expectativas do teste da proposta de melhoria, correspondem às variáveis dependentes.

Para diminuir a necessidade de interagir diretamente com estes conceitos, a atividade foi adaptada para que o executor apenas se preocupe em identificar as métricas que possam mensurar os efeitos da mudança e quantificar a influência de outros fatores nos resultados da avaliação.

#### E) Escolher o *Design* => Descrever Melhoria para Avaliação

A escolha do *Design* está relacionada à quantidade de fatores e tratamentos a serem aplicados no decorrer do projeto-piloto. Esta atividade do processo original foi reduzida à declaração das ações de melhoria aplicadas ao processo, ou seja, no registro do tratamento a ser aplicado ao fator processo. Portanto, com relação à experimentação, o design predefinido para o teste da proposta de melhoria se reduz a **um fator com um tratamento**. Vale ressaltar que não há interesse em comparar uma proposta de melhoria com outras, apenas deseja-se testar se os efeitos observados correspondem às expectativas, o que diminui significativamente a complexidade do estudo.

#### F) Determinar Instrumentação => Preparar para Executar

Os instrumentos necessários à condução do piloto são constituídos por ferramentas que compõem o ambiente de desenvolvimento de software da organização e os elementos da própria abordagem. A instrumentação também envolve a tecnologia que será testada e materiais de apoio à sua utilização.

As providências para disponibilizar estes instrumentos durante a execução do projeto-piloto é uma atividade inerente ao planejamento do projeto e preparação do ambiente de execução. Assim, a atividade de determinação da instrumentação foi concatenada à atividade que trata da preparação para execução do projeto-piloto.

#### G) Tratar a Validade => Identificar Fatores de Influência + Identificar Piloto + Elaborar Plano de Medição

Um experimento científico precisa tratar meticulosamente questões relacionadas à validade, ou seja, questões que definem a acurácia e o quão geral será o resultado obtido. No contexto de aplicação deste processo, o escopo é reduzido ao âmbito organizacional. Este fato simplifica o tratamento da validade, que será focado na determinação de quantos projetos serão necessários para avaliar a relação causa-efeito da proposta de melhoria e na identificação e mensuração da influência dos fatores de confusão na análise dos efeitos provocados pelas ações que estão sendo testadas.

Assim, a atividade original foi fracionada em três: no momento da identificação do piloto, estimam-se quantos projetos serão necessários para avaliar os efeitos da proposta que será testada; na elaboração do plano de medição as variáveis independentes que podem exercer alguma influência no teste das melhorias são selecionadas; e na identificação dos fatores de influência, os riscos de execução do piloto que podem causar distorção aos dados coletados são enumerados e estabelecidas diretrizes para mitigação e contingenciamento.

#### H) Preparar para Execução

A preparação da execução do piloto confunde-se com a atividade de preparação para execução do projeto. A primeira está relacionada com a preparação necessária para que a mudança do processo seja aplicada adequadamente no projeto, enquanto que a segunda consiste em prover todas as condições necessárias para o sucesso do projeto.

As necessidades de treinamento, adequações no ambiente e demais atividades de preparação devem ser executadas considerando os objetivos do projeto e deverá ser assegurado que as mudanças introduzidas no processo do projeto sejam assimiladas pela equipe de forma a contribuir com o sucesso do projeto.

Esta atividade limita-se a evocar as atividades de planejamento, treinamento e preparação para o projeto. Como insumo, a atividade recebe um documento de diretrizes para ser utilizado durante as atividades de preparação para execução do projeto.

#### I) Execução => Executar Proposta de Melhoria + Monitorar Projetos-Piloto

Durante a execução dos projetos-piloto é realizada a atividade de monitoração para acompanhar o andamento do projeto à medida que as ações de melhoria estão sendo executadas, pois dependendo do piloto, mesmo que o projeto não seja concluído na íntegra, a avaliação dos efeitos da mudança no processo poderá ser realizada. Assim, identificar de forma não intrusiva o momento que as ações de melhoria estão sendo executadas, bem como aqueles que antecedem e sucedem à execução das ações, pode ser muito útil à condução do piloto, propiciando condições favoráveis à observação e captura de outras informações relevantes à análise qualitativa dos dados.

#### J) Mensurar => Coletar Medidas

Apenas o nome da atividade foi alterado.

#### K) Validar Dados => Conferir Medição dos Projetos-Piloto + Realizar Reunião de Avaliação

Sem a devida validação dos dados os resultados do piloto podem ser distorcidos. Atenção especial deve ser dada aos procedimentos adotados para coleta e ao momento em que cada métrica foi coletada, pois a postergação da coleta de uma medida poderá acarretar na imprecisão do registro. Além disso, uma reunião com a equipe pode auxiliar a revelar inconsistências e desvio nos dados, bem como subsidiar a análise qualitativa destes dados a partir de uma avaliação considerando a perspectiva da equipe do projeto.

#### L) Adequar Dados à Análise + Teste de Hipóteses => Análise Quantitativa.

Identificar medidas que se distanciam excessivamente da média (*outliers*), tratar os dados para torná-los passíveis de comparação e aptos a serem submetidos aos testes de hipóteses são atividades que exigem conhecimento específico em estatística, como por exemplo, a identificação do teste mais adequado ao contexto. Além disso, um fator complicador para a indústria de software é que a execução de pilotos não é capaz de gerar dados suficientes para subsidiar a execução dessas atividades.

Diante deste fato, a atividade simplificada passou a consistir na realização da compilação dos dados coletados a partir dos projetos-piloto e verificação dos resultados com relação às expectativas declaradas para o piloto.

#### M) Emitir Conclusões => Ratificar Fatores de Influência + Análise Qualitativa

No processo original, as conclusões são fundamentadas nos resultados da verificação das hipóteses e dos dados coletados durante a realização do estudo. Em função da ausência de controle observada no ambiente industrial, os fatores de confusão que foram identificados a priori podem ou não terem exercido influência, bem como outros fatores podem ter sido identificados apenas durante a execução dos projetos-piloto. Então, antes de analisar qualitativamente, é preciso investigar a ação destes fatores sobre cada projeto.

Na análise qualitativa, a análise quantitativa, a influência dos fatores e a opinião dos participantes são consideradas para determinar a efetividade da proposta de melhoria que está sendo testada.

#### N) Formatar Resultados + Emitir Relatório Final => Emitir Laudo de Avaliação do Piloto

O plano do piloto, os resultados quantitativo e qualitativo e todos os fatores que influenciaram o piloto são concatenados e devidamente formatados para que fique

inteligível a todos os interessados. Essa formatação é realizada sobre o documento Laudo de Avaliação do Piloto.

#### O) Registrar Lições Aprendidas

Planos, diretrizes, artefatos gerados e sugestões de melhorias ao próprio processo para realização de pilotos são ativos que devem compor a memória organizacional. Estes ativos devem ser depositados na base de conhecimento organizacional e devem ficar disponíveis, pois representam uma rica fonte de conhecimento. Além disso, a execução de pilotos é uma ferramenta valiosa no que tange a explicitação do conhecimento e no auto-conhecimento organizacional, ou seja, é possível capturar e compreender melhor os fatores que influenciam no comportamento da organização.

Vale ressaltar que na adaptação das atividades do processo original também foi dada ênfase no vocabulário, para que a interpretação do processo se tornasse tão inteligível quanto possível.

### **5.3.3. Modelo do processo**

Quando há necessidade de avaliar a efetividade de determinada proposta de melhoria, esta é submetida à abordagem. Este evento é responsável pelo início do processo de realização de pilotos que culmina com a emissão do Laudo de Avaliação do Piloto (Figura 5.1).

Vale ressaltar que a decisão de realizar a avaliação da efetividade precede a avaliação propriamente dita. Avaliações envolvem custo e esforço, portanto deve-se ter em mente que existem alternativas e estas devem ser consideradas. Por exemplo, prover um julgamento coletivo da proposta envolvendo uma equipe experiente ou optar pela institucionalização e tratar as mudanças com uma abordagem de gerenciamento de riscos. Sugere-se também não optar pela realização de pilotos para avaliar questões de "vida curta", ou seja, mudanças temporárias que tratam questões específicas cujo descarte já está previsto. Este modelo de processo assume que estas questões já foram consideradas e que a decisão de verificar se a proposta de melhoria é efetiva através de um piloto foi tomada.

A proposta de melhoria precisa conter os objetivos de melhoria e sua relação com os objetivos organizacionais, os resultados esperados com a aplicação das ações de melhoria no(s) processo(s), a descrição detalhada das ações de melhoria e o contexto ao qual as melhorias se aplicam (tipo, tamanho e duração de projeto, paradigma, etc.). Uma vez que a proposta atenda os critérios, o processo é iniciado e

a primeira atividade a ser executada é a que trata da definição do piloto a ser realizado.

O processo é finalizado quando todos os projetos-piloto selecionados para testar a proposta são concluídos e os dados coletados são compilados através de uma análise quantitativa seguida de uma análise qualitativa. No final do processo o laudo de avaliação do piloto é elaborado contendo os dados consolidados, as ponderações realizadas durante a análise e a informação se os resultados atenderam ou não às expectativas.

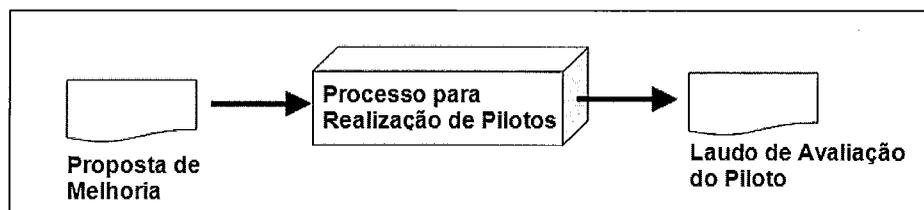


Figura 5.1 - Visão macro do processo

A Figura 5.2 apresenta a representação gráfica do modelo do processo para realização de pilotos utilizando a linguagem para modelagem de processos proposta por Vilela (2004) (o Anexo 1 lista os elementos que compõem esta linguagem). O modelo envolve a participação dos seguintes papéis:

**Grupo de Processo:** Pessoas responsáveis pela definição e melhoria dos processos e ativos de processo da organização;

**Gerente do Projeto:** Responsável pela condução das atividades dos projetos desenvolvidos pela organização;

**Equipe do Projeto:** Pessoas que participam do projeto. Dentre eles: analistas, testadores, programadores, arquitetos de software e projetistas.

As atividades do modelo, oriundas das adaptações realizadas nas atividades do processo experimental, foram agrupadas em nove macro-atividades: Definir piloto, definir diretrizes para realização do piloto, elaborar plano do piloto e documento com diretrizes para realização do piloto, preparar para executar projeto-piloto, monitorar piloto, analisar resultados dos projetos-piloto, visualizar resultados e registrar lições aprendidas. A seguir, serão apontadas questões chaves para realização de cada uma das macro-atividades do modelo. O modelo do processo pode ser encontrado em detalhes no Apêndice G desta dissertação.

#### 5.3.3.1. Definir piloto

A primeira macro-atividade consiste na identificação do piloto, descrição das melhorias a serem avaliadas, definição de um plano de medição e declaração das expectativas para o projeto piloto.

Durante a identificação do piloto, o grupo de processos da organização deve procurar analisar a abrangência da proposta e verificar a dificuldade para avaliar os efeitos desejados. Quanto mais difícil de observar os efeitos, maior deve ser o número de projetos que deverão ser associados ao piloto.

A descrição das ações a serem testadas no processo deve ser realizada tomando como referência o processo ao qual a proposta se aplica. Estas melhorias podem consistir na alteração de atividades (inclusão, alteração, exclusão ou reordenação), utilização de uma nova ferramenta de apoio, utilização de novos métodos e/ou técnicas na execução de determinadas atividades, alteração ou inserção de novos modelos e diretrizes. Deve-se ter atenção para aquelas ações independentes que podem ter influência mútua, principalmente quando a proposta visa atender mais de um objetivo de melhoria. As ações que se influenciam mutuamente podem dificultar a observação dos reais efeitos que cada uma provocou no processo.

Para demonstrar melhorias no processo de software ele precisa ser mensurado (MORISIO, 1999). Métricas de processos e produto são usadas para responder se esforços de melhoria são efetivos (WALRAD; MOSS, 1993). A partir da definição dos objetivos, através do método GQM, é possível identificar quais as questões e métricas que devem ser utilizadas para mensurar os efeitos das alterações do processo (VAN SOLINGEN; BERGHOUT, 1999).

A declaração das expectativas deve ser expressiva e, ao mesmo tempo, exeqüível. Expectativas muito altas são difíceis de serem satisfeitas, e isso pode implicar em uma avaliação quantitativa negativa e ocultar a efetividade de uma proposta. Por outro lado, expectativas muito baixas não são úteis, pois, de acordo com o tipo de melhoria, o efeito positivo observado quantitativamente pode não ser expressivo o suficiente para justificar a institucionalização das melhorias. Organizações imaturas têm dificuldade para declarar expectativas, por mais simples que sejam. Sugerem-se como importantes fontes: *benchmarking*, análise de dados históricos e opinião de especialistas.

#### 5.3.3.2. Definir diretrizes para realização do piloto

Esta macro-atividade resultará nas diretrizes que apoiarão a seleção do projeto a ser utilizado como piloto para avaliar a melhoria. Também está inclusa a atividade

que apoiará a identificação dos fatores que podem exercer alguma influência durante a execução do projeto, de forma a dificultar a avaliação quantitativa e/ou qualitativa das melhorias a serem testadas.

Há casos onde a avaliação somente poderá ser realizada se o projeto apresentar uma determinada característica. Por exemplo: se no processo de desenvolvimento de software foi inserida uma atividade que se aplica apenas ao desenvolvimento de software para web, o projeto-piloto deverá possuir tal característica. Da mesma forma, caso a organização queira avaliar o desempenho do grupo de trabalho com o uso de uma nova tecnologia, é necessário selecionar participantes que tenham o mesmo nível de conhecimento da tecnologia anterior.

#### 5.3.3.3. Elaborar plano do piloto e documento com diretrizes

O plano do piloto deve conter todos os dados necessários à condução da avaliação da proposta de melhoria. Em síntese, ele concatena todos os artefatos produzidos pelas atividades de definição do piloto e das diretrizes para realização do piloto. O documento que contém as diretrizes provê um roteiro para que o gerente do projeto verifique a adequação do projeto ao piloto e insira as mudanças no processo do projeto. As diretrizes também fornecem orientações quanto à preparação do ambiente de operação e condução do projeto que servirá de piloto.

#### 5.3.3.4. Preparar para execução

O grupo de processo encaminha o documento de diretrizes para realização do piloto para o gerente do projeto. O gerente do projeto trata de associar o seu projeto ao piloto e utiliza as diretrizes para planejar o processo, planejar o projeto, treinar a equipe, estabelecer o plano de medição e preparar o ambiente para execução do projeto. É importante ressaltar que estas tarefas associadas à preparação também podem pertencer ao escopo de outros processos e/ou áreas de processo (vide Figura 5.3).

#### 5.3.3.5. Executar projetos-piloto

A equipe do projeto executa as atividades previstas no plano do processo, inclusive as que constituem a proposta de melhoria. Em seguida, ou concomitantemente, as medidas são coletadas e depositadas na base de medições.

#### 5.3.3.6. Monitorar projetos-piloto

Dependendo do tipo de proposta que está sendo testada, pode não ser necessário finalizar os projetos para realizar a análise dos dados coletados durante a execução das partes que foram alteradas no processo. Ao realizar uma monitoração

freqüente, além de identificar a partir de qual momento o projeto-piloto pode ser dado como finalizado, a monitoração pode ser um instrumento que permita ao grupo de processo observar de perto o momento em que a ação de melhoria irá ser executada e, assim, coletar outras informações que irão apoiar a análise. Contudo, recomenda-se realizar a monitoração de forma não intrusiva para diminuir potenciais influências associadas ao sentimento de “sob avaliação” durante a execução do projeto.

Enquanto o grupo de processos monitora a execução das propostas que estão sendo testadas, a equipe do projeto também realiza o registro da medição para as métricas do projeto-piloto.

Antes das atividades de análise dos dados, quando o projeto-piloto for considerado finalizado, realiza-se uma reunião para capturar a avaliação da proposta sob a perspectiva dos participantes. Da mesma forma que, durante a execução de experimentos, o *feedback* dos participantes contribui com a validação dos dados, na verificação da conformidade do processo, na identificação de problemas de condução e no entendimento da percepção dos participantes (KARAHASANOVIC et al., 2005) recomenda-se que a análise qualitativa também faça uso deste recurso.

A dificuldade para executar a proposta, resistência de alguns participantes da equipe, dentre outros aspectos, pode auxiliar de maneira significativa a análise qualitativa dos efeitos observados.

#### 5.3.3.7. Analisar resultados dos projetos-piloto

Os dados coletados são computados e comparados às expectativas declaradas na proposta de melhoria. O resultado da análise quantitativa poderá atender integralmente, parcialmente ou não atender às expectativas. Quando mais de um projeto está associado ao piloto, é possível obter resultados conflitantes, ou seja, o atendimento parcial às expectativas. Nestes casos, a análise qualitativa deve ser ainda mais criteriosa. O grupo de processo deve examinar com cuidado os fatores de influência e o impacto de cada um destes fatores nos projetos e considerar os dados coletados durante a monitoração e na reunião de *feedback* para apoiar a interpretação dos resultados.

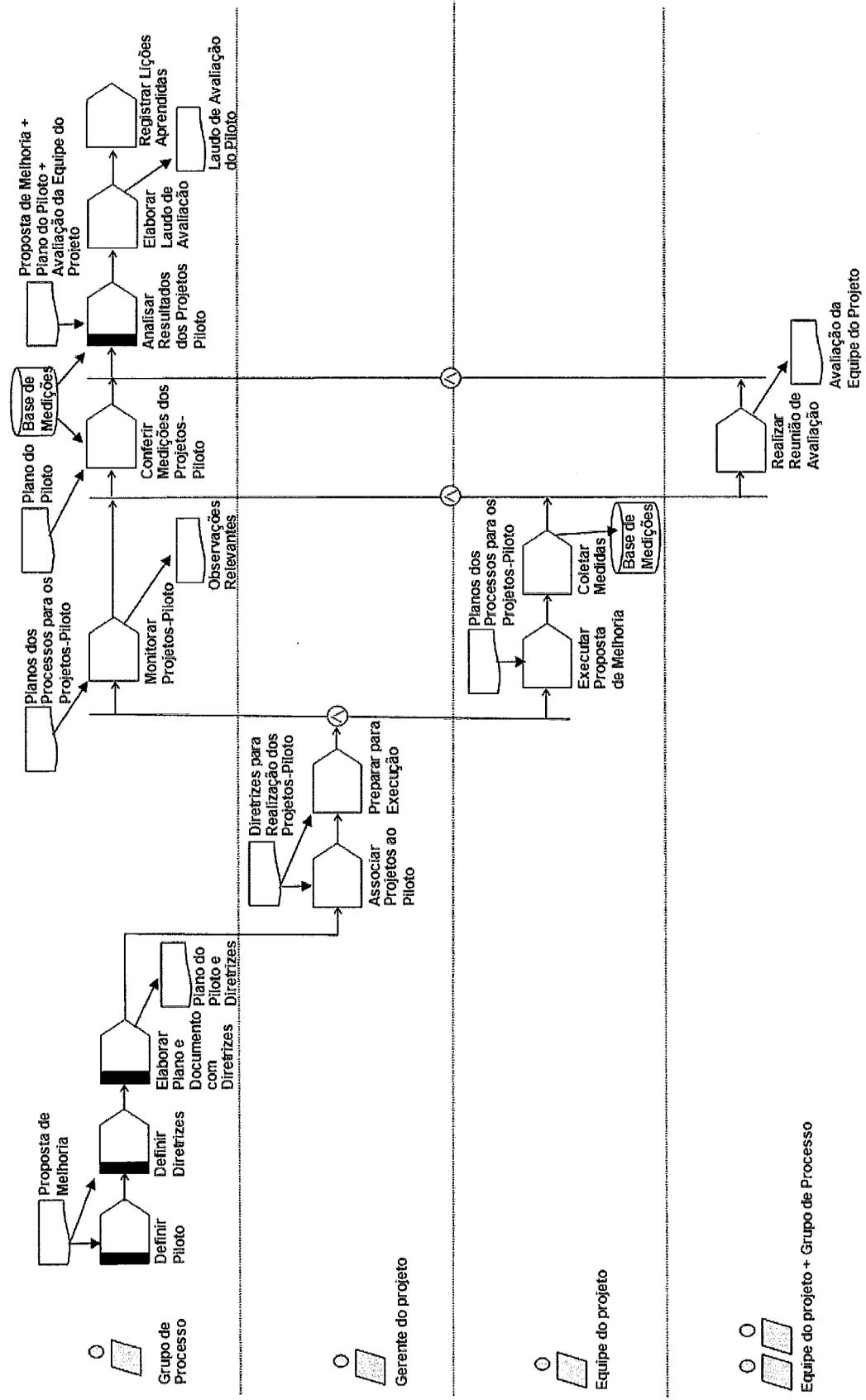


Figura 5.2 - Modelo do Processo para Realização de Pilotos

### 5.3.3.8. Finalizar Piloto

Todos os dados referentes ao piloto são concatenados e formatados em um único relatório que ao final apresenta o parecer a respeito da avaliação da efetividade da proposta de melhoria (Laudo de Avaliação do Piloto)

Além disso, todos os artefatos gerados no decorrer da realização do piloto devem ser incorporados à base de ativos da organização. Estes artefatos podem prover conhecimento para apoiar novas avaliações, servir como referência para futuras tomadas de decisão, fornecer indícios sobre a relação entre aspectos organizacionais e até motivar novas experiências em prol da melhoria dos processos.

A partir dos registros realizados durante a monitoração dos projetos-piloto, do laudo da avaliação e dos dados coletados durante a reunião com a equipe que participou no piloto, são extraídas as lições aprendidas com a experiência.

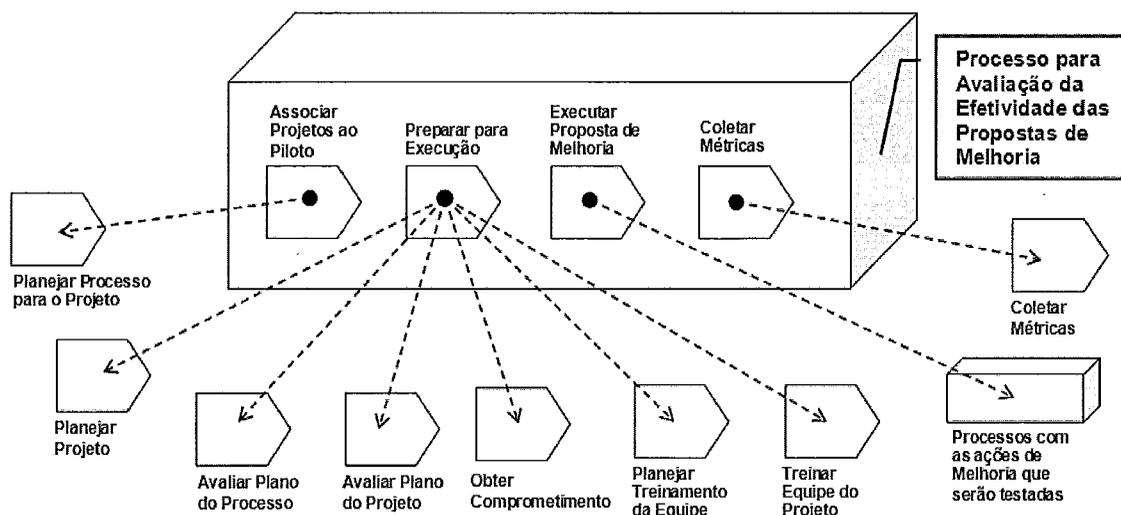


Figura 5.3 - Interação do processo com outras atividades e processos da organização

## 5.4. A Ferramenta Pilot

O estudo apresentado em Silva Filho et al. (2005) apresentou um custo de aproximadamente 3% do projeto que foi utilizado como piloto. Tendo em vista as atividades que foram realizadas, espera-se que com apoio ferramental o esforço possa ser reduzido significativamente, propiciando inclusive a associação de mais de um projeto ao mesmo piloto sem que isso tenha impacto significativo nos custos do estudo.

A ferramenta **Pilot**, definida e implementada como parte desta dissertação, automatiza a execução do elemento processo da abordagem. A ferramenta não dá apoio a todas as atividades do processo. Foram selecionadas apenas atividades onde a automação traria mais benefícios, considerando principalmente os aspectos esforço

e conhecimento requerido. A Figura 5.4 ilustra as atividades do processo que foram contempladas na ferramenta. No capítulo 6, a ferramenta **Pilot** é descrita em detalhes.

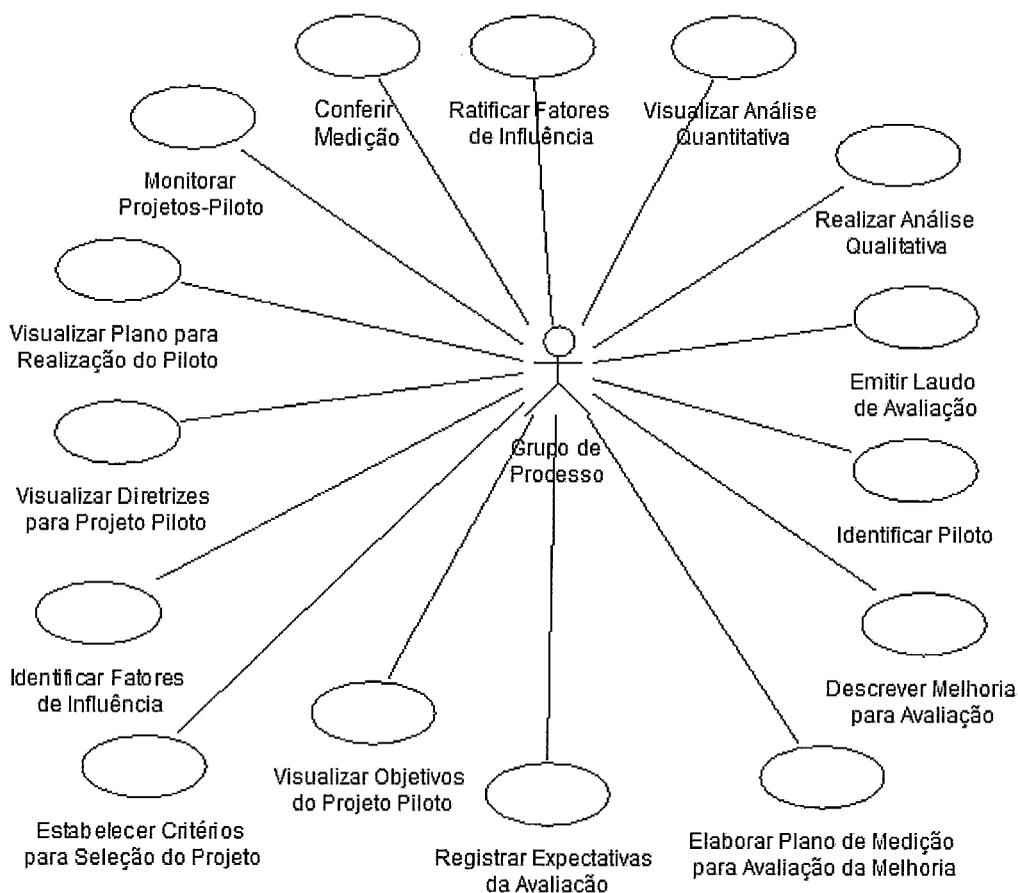


Figura 5.4 - Diagrama de casos de uso com as funcionalidades da Pilot.

## 5.5. O Guia

O objetivo do guia é fornecer o conhecimento necessário para compreensão dos elementos da abordagem e habilitar os usuários a utilizá-la. Está dividido em três seções: a primeira apresenta uma visão geral e informa como o guia deve ser utilizado; a segunda parte do guia contém um roteiro para aplicação do processo através da ferramenta, partindo de um exemplo real e uso de um vocabulário adequado ao perfil do grupo de processo; e a última parte contém diretrizes e informações que apóiam o usuário a tomar decisões e sensibilizá-lo ao contexto da avaliação para que ele seja capaz de capturar informações importantes ao longo da realização do piloto e utilizá-las na análise qualitativa.

Dentre as questões centrais abordadas pelo guia destacam-se:

### A) Número de projetos-piloto

Fornece subsídios para que o usuário possa identificar quantos projetos-pilotos são necessários para avaliar uma proposta de melhoria, com base no risco, amplitude da proposta, urgência, disponibilidade de recursos e impacto na organização.

## B) Tratamento da validade

A questão da validade está relacionada aos conceitos de validade interna e externa abordados por Wohlin et al. (2000), que tratam, respectivamente, da qualidade dos dados coletados e do significado destes dados com relação ao contexto geral (potencial de generalização). Considerar contexto, critérios de seleção do projeto a ser utilizado como piloto, seleção dos indivíduos, necessidade de treinamento, e fatores que podem influenciar a avaliação dos resultados é indispensável.

## C) Diretrizes para seleção de métricas

O GQM facilita a identificação de métricas para mensuração dos efeitos da mudança, entretanto também é necessário mensurar os fatores que podem influenciar nos efeitos que se pretende observar. Por outro lado, mensurar todos os fatores pode comprometer a medição e implicar no aumento do consumo de recursos do projeto.

## D) Identificação dos fatores de influência

O ambiente industrial está sujeito a intempéries de todos os tipos. É importante concentrar-se nos riscos que podem afetar diretamente a proposta. Tão importante quanto prever, é identificar quando o evento ocorreu, pois mesmo que não seja possível remediar os efeitos da ocorrência do risco, a análise qualitativa deve levá-los em consideração durante a confecção do laudo.

## E) Realizar reunião de avaliação

Para conduzir reuniões com alto grau de aproveitamento podem ser aplicadas técnicas específicas, conforme o objetivo da reunião. O guia apresenta algumas alternativas para dinamizar e potencializar a coleta do *feedback* dos participantes dos projetos-piloto.

## F) Análise qualitativa

A análise qualitativa envolve muita subjetividade. É importante que o grupo de processo tenha sensibilidade ao avaliar o contexto ao qual cada projeto-piloto foi realizado e considerar estas informações junto com os dados quantitativos. O estudo das relações causa-efeito requer atenção, percepção e capacidade de identificar relações entre aspectos que, a priori, não aparentavam possuir relação.

## 5.6. Considerações Finais

Este capítulo apresentou a abordagem baseada em experimentação para avaliar a efetividade das propostas de melhoria de processos de software através do uso de pilotos.

Os três elementos que compõe a abordagem provêm sistematização, automação e conhecimento para a realização de pilotos.

A abordagem tende a potencializar a realização das avaliações das propostas de melhoria, tornando-as mais acuradas, sem que isso implique em maiores custos ou aumento do esforço. No capítulo 6 esta questão é retomada e um piloto é proposto para avaliar as contribuições da abordagem.

Através da utilização da abordagem, as organizações que adotam e/ou têm interesse em obter níveis superiores em avaliações com relação aos modelos de qualidade CMMI e MPS.BR poderão contar com a ampla cobertura das práticas e o fornecimento de indicadores relacionados à avaliação de propostas de melhorias e realização de pilotos sugeridos por estes modelos.

## **Capítulo 6. Pilot, Uma Ferramenta para Apoiar a Realização de Pilotos**

---

### **6.1. Introdução**

Neste capítulo serão apresentados os aspectos mais relevantes do desenvolvimento da ferramenta Pilot, um dos elementos da abordagem, dentre eles: o planejamento inicial, os requisitos, a modelagem e o desenvolvimento da interface com o usuário, bem como questões relacionadas à integração da ferramenta ao ambiente de engenharia de software centrado em processo, neste caso, a Estação TABA.

Em seguida, será feito um relato sobre um piloto da abordagem com o intuito de avaliar o esforço requerido para sua aplicação e sua utilidade em um ambiente industrial.

Por fim, considerações serão realizadas sobre o desenvolvimento e utilização da ferramenta no âmbito industrial.

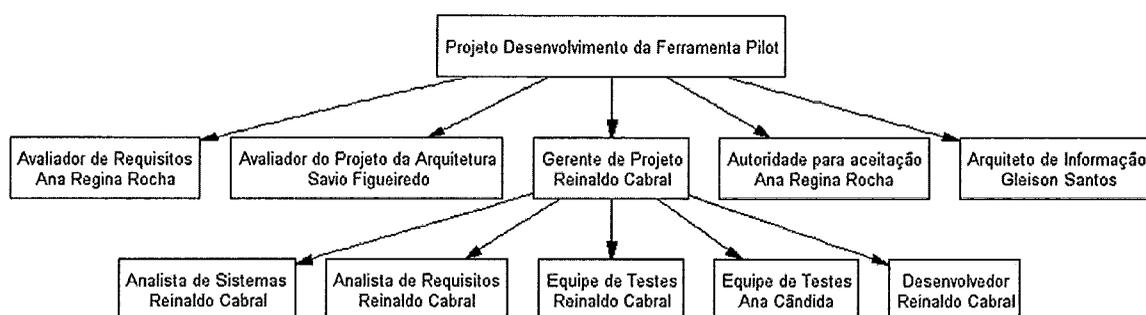
### **6.2. O Processo de Desenvolvimento**

#### **6.2.1. Planejamento inicial**

No planejamento inicial foi realizado um conjunto de atividades com o intuito de caracterizar o projeto, determinar o modelo de ciclo de vida mais adequado, elaborar o planejamento do processo e elaborar a versão preliminar do plano do projeto.

O ciclo de vida escolhido para o projeto foi o cascata, conforme as características do projeto e, especialmente, a estabilidade dos requisitos, a não necessidade de entrega de versões intermediárias e a inexistência de restrições de prazo.

Uma vez que o domínio de aplicação é bastante específico, foi necessário incluir no processo de desenvolvimento atividades de prototipação para facilitar a validação dos requisitos elicitados e verificar o atendimento aos requisitos de usabilidade impostos pelo ambiente ao qual a ferramenta será integrada.



**Figura 6.1 - Estrutura Organizacional do Projeto**

A estrutura organizacional do projeto foi planejada e envolveu a alocação de pessoas da equipe do Projeto TABA aos papéis.

**Tabela 6.1 - Outras partes envolvidas no projeto.**

Papéis	Responsáveis
Fornecedor de Requisitos	Ana Regina Rocha e Reinaldo Cabral
Fornecedor de Requisitos do Sistema	Gleison Santos
GQPP	Mariano Montoni
Grupo de Gerência de Configuração	Gleison Santos

Na primeira versão do cronograma, elaborada durante o planejamento inicial, foi estimado um total de 1004 horas para realização do projeto, com a data de início em 16 de maio de 2005 e previsão de término para o dia 07 de novembro do mesmo ano (Tabela 6.2).

**Tabela 6.2 - Cronograma inicial.**

Macro atividades	Duração Prevista	Data Inicial	Data Final	Recurso Alocado
Planejamento do Processo para o Projeto	8h	16/05/05	16/05/05	Reinaldo Cabral
Elicitação de Requisitos	24h	17/05/05	19/05/05	Reinaldo Cabral
Planejamento do Projeto	8h	20/05/05	20/05/05	Reinaldo Cabral
Análise dos Requisitos do Sistema	32h	23/05/05	26/05/05	Reinaldo Cabral
Projeto da Arquitetura do Sistema	16h	27/05/05	30/05/05	Reinaldo Cabral
Análise dos Requisitos do Software	32h	31/05/05	03/06/05	Reinaldo Cabral
Projeto do software	100h	06/06/05	22/06/05	Reinaldo Cabral
Construção do Software	300h	22/06/05	12/08/05	Reinaldo Cabral
Integração do software	80h	15/08/05	26/08/05	Reinaldo Cabral
Teste do Software	160h	29/08/05	23/09/05	Reinaldo Cabral
Integração do Sistema	72h	26/09/05	07/10/05	Reinaldo Cabral
Teste do Sistema	100h	07/10/05	25/10/05	Reinaldo Cabral
Implantação do software	24h	25/10/05	28/10/05	Reinaldo Cabral
Apoio à aceitação do software	32h	28/10/05	03/11/05	Reinaldo Cabral
Encerramento do Desenvolvimento	16h	03/11/05	07/11/05	Reinaldo Cabral

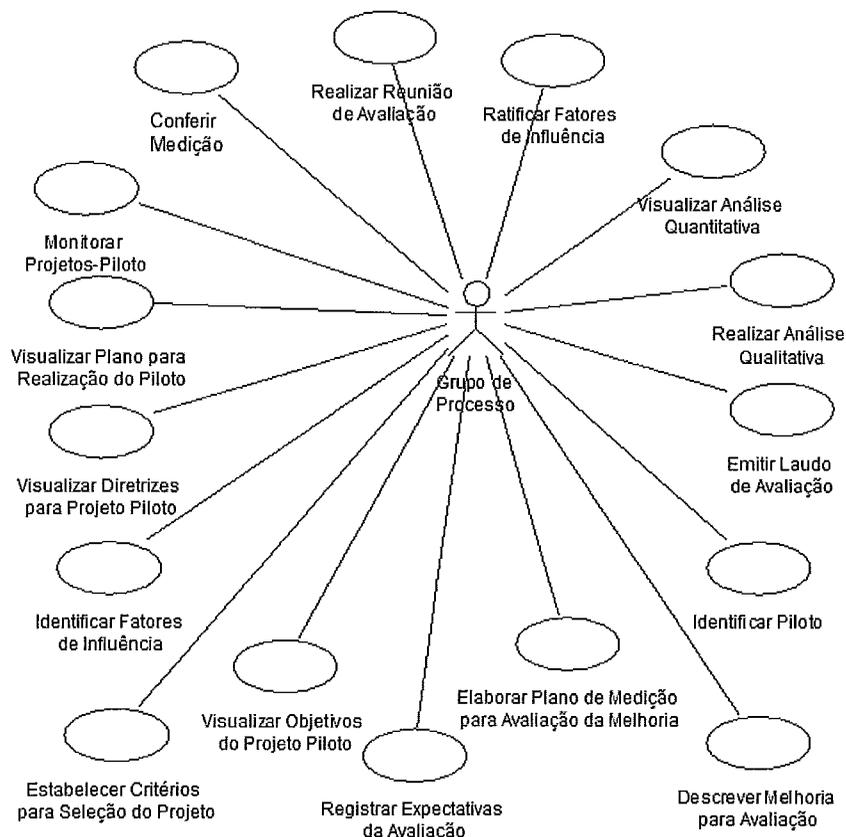
O plano de custo tratou apenas de enumerar as pessoas envolvidas, visto que o desenvolvimento deste projeto não envolveu desembolso financeiro de qualquer parte. O plano de custos também enumerou os recursos de hardware, software e infraestrutura necessária ao desenvolvimento do projeto, bem como as respectivas datas para a disponibilidade do recurso (Tabela 6.3).

**Tabela 6.3 - Disponibilidade de recursos para o projeto.**

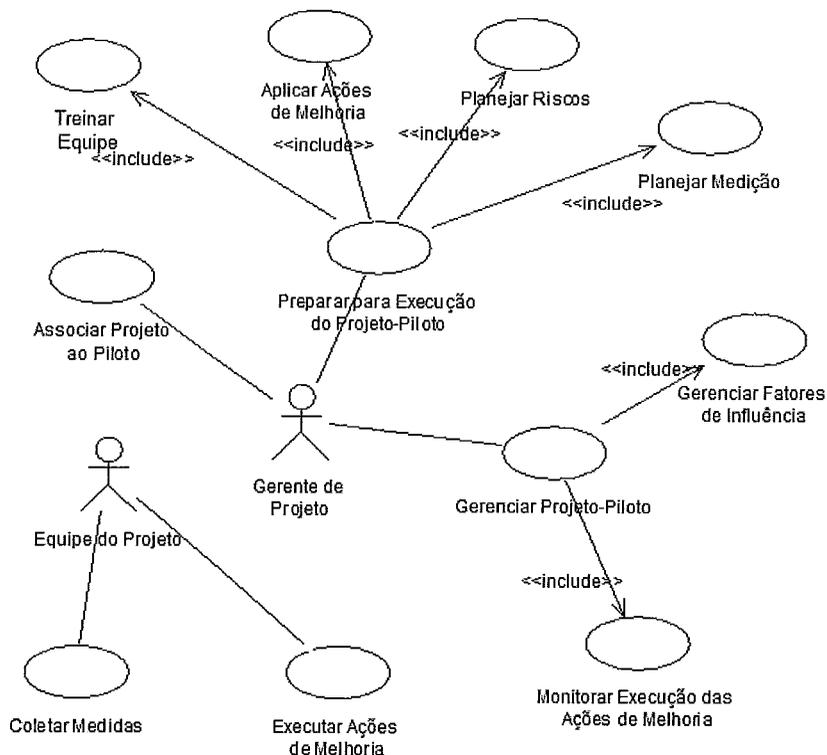
Recursos de Hardware		
Nome	Quantidade	Disponível em
Computador (Estação de Trabalho)	1	16/05/2005
Impressora	1	16/05/2005
Computador (Servidor CVS)	1	16/05/2005
Recursos de Software		
Nome	Quantidade	Disponível em
Rational Rose 2000 Enterprise Edition	1	23/05/2005
MS Visual C++ 6.0	1	22/06/2005
MS SQL Server 2000	1	22/06/2005
EditPlus 1.0	1	22/06/2005
MS Word 2000	1	16/05/2005
WinCVS 1.3	1	16/05/2005
Estação TABA	1	16/05/2005
Ms Project	1	16/05/2005
MS Excel	1	16/05/2005
Infra-estrutura		
Nome	Quantidade	Disponível em
Modem ADSL 256kbps	1	16/05/2005
Assinatura Provedor ADSL	1	16/05/2005
Linha telefônica para conexão ADSL	1	16/05/2005

## 6.2.2. Requisitos

A principal fonte dos requisitos foi o próprio processo para realização de pilotos. A partir do processo, todas as funcionalidades potenciais para a ferramenta foram identificadas.



**Figura 6.2 - Casos de uso representando a interação do grupo de processo com a abordagem**



**Figura 6.3 - Casos de uso representando a interação do gerente e equipe do projeto com a abordagem**

Para selecionar quais os requisitos que seriam contemplados pela ferramenta, os requisitos foram priorizados a partir de critérios pré-estabelecidos, dentre eles: complexidade de execução, esforço requerido, existência de ferramentas alternativas. Assim, apenas os casos de uso envolvidos na interação do grupo do processo com a abordagem (Figura 6.2) foram selecionados para compor as funcionalidades da ferramenta Pilot.

Para apoiar a avaliação dos requisitos funcionais e de usabilidade foi construído um protótipo para apresentar a disposição das funcionalidades na ferramenta, para ilustrar a navegabilidade e demonstrar a interação do usuário com os recursos oferecidos pela ferramenta Pilot. As figuras 6.4 e 6.5 representam, respectivamente, a versão inicial do protótipo e versão final da tela Descrever Melhorias para Avaliação. As principais diferenças entre estas versões são: (i) Redução significativa da necessidade de digitação de dados pelo usuário. A evolução da interface permitiu a interação direta do usuário com o processo a ser melhorado, evitando erros em decorrência da digitação; (ii) Cômputo automático das ações de melhoria no processo operações. Na versão inicial, as ações de melhoria tinham que ser registradas manualmente pelo usuário; (iii) Tela única para inclusão, alteração e exclusão de atividades; (iv) Inclusão da possibilidade de registrar melhorias não associadas às atividades específicas do processo.

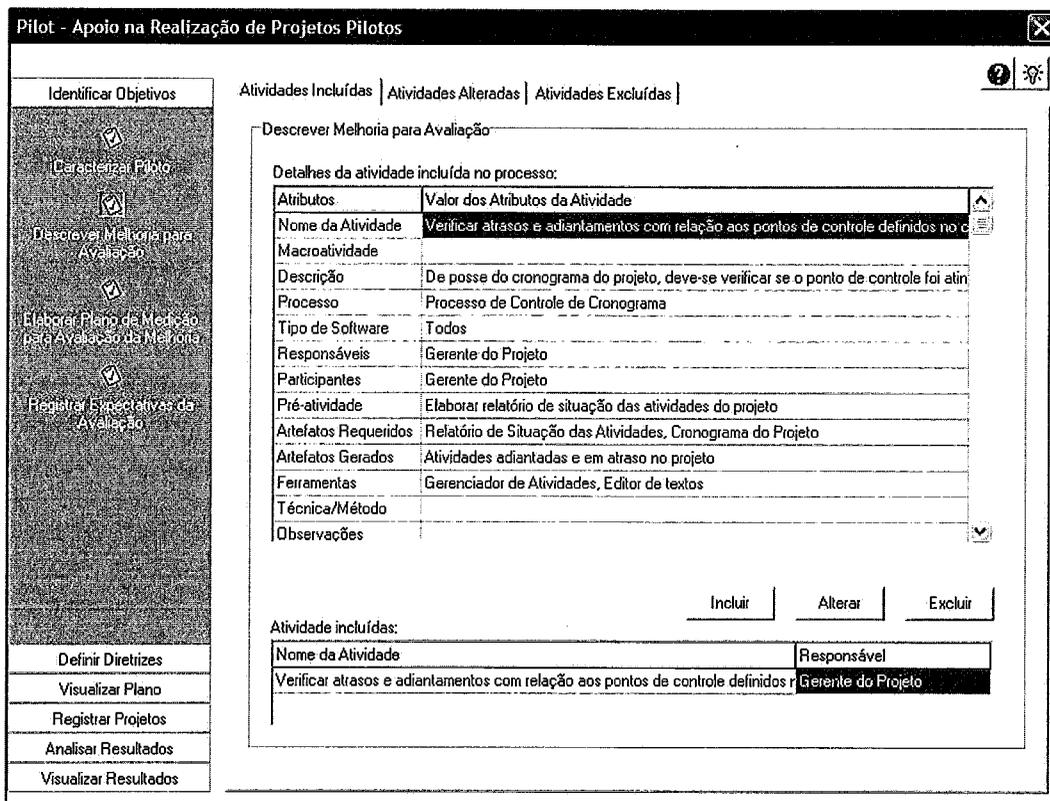


Figura 6.4 - Protótipo da tela para descrever as ações de melhoria.

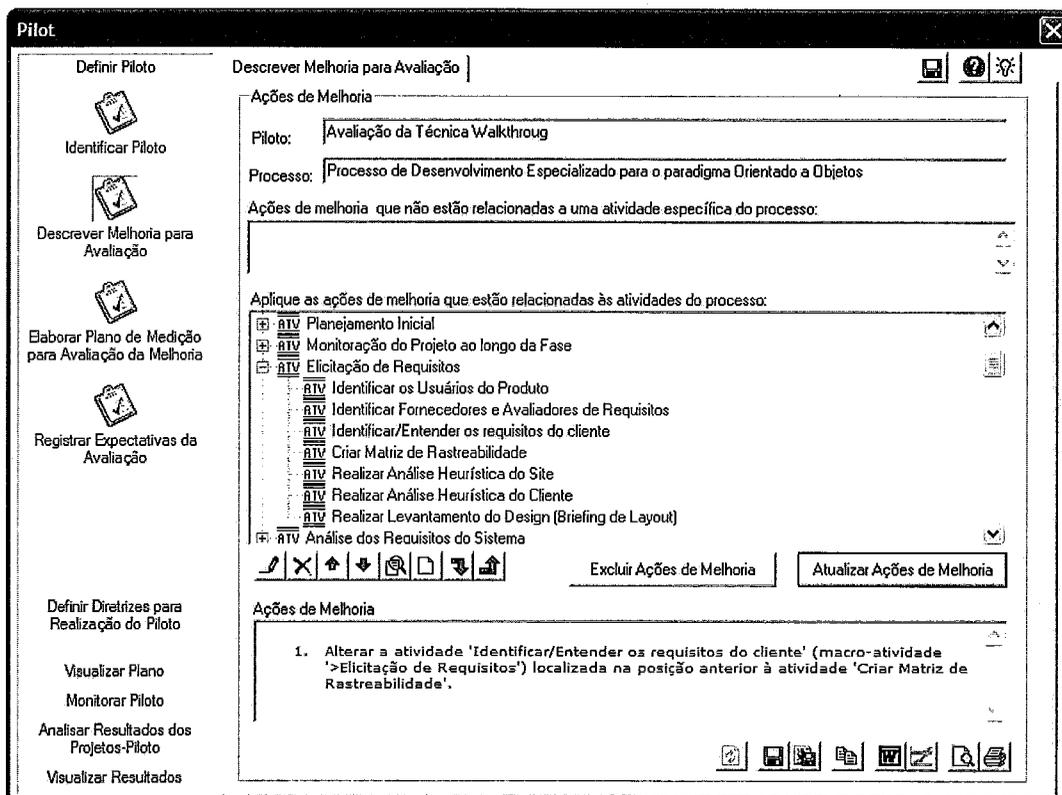


Figura 6.5 - Versão corrente da tela para descrição das ações de melhoria.

### 6.2.3. Projeto

A partir dos requisitos e da análise dos componentes de domínio fornecidos pela Estação TABA o diagrama de classes foi elaborado (Figura 6.8). Até a aprovação do modelo, foram geradas 15 versões preliminares.

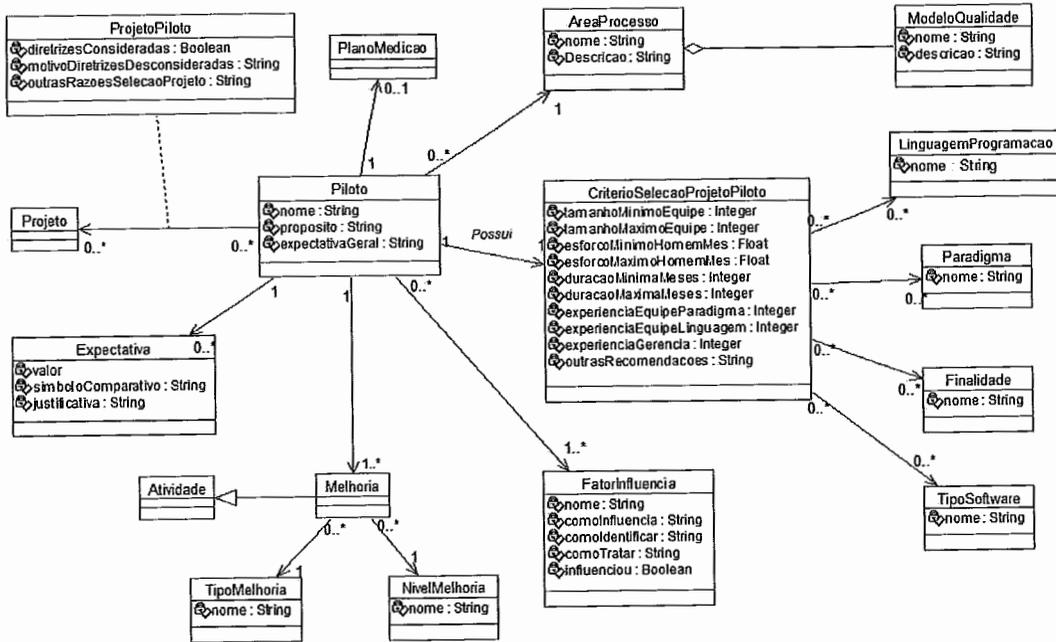


Figura 6.6 - Primeira versão do diagrama de classes.

Diagramas de estados também foram elaborados para que fosse possível prover mecanismos de apoio à monitoração do piloto (Figura 6.7) e seus respectivos projetos-pilotos (Figura 6.9).

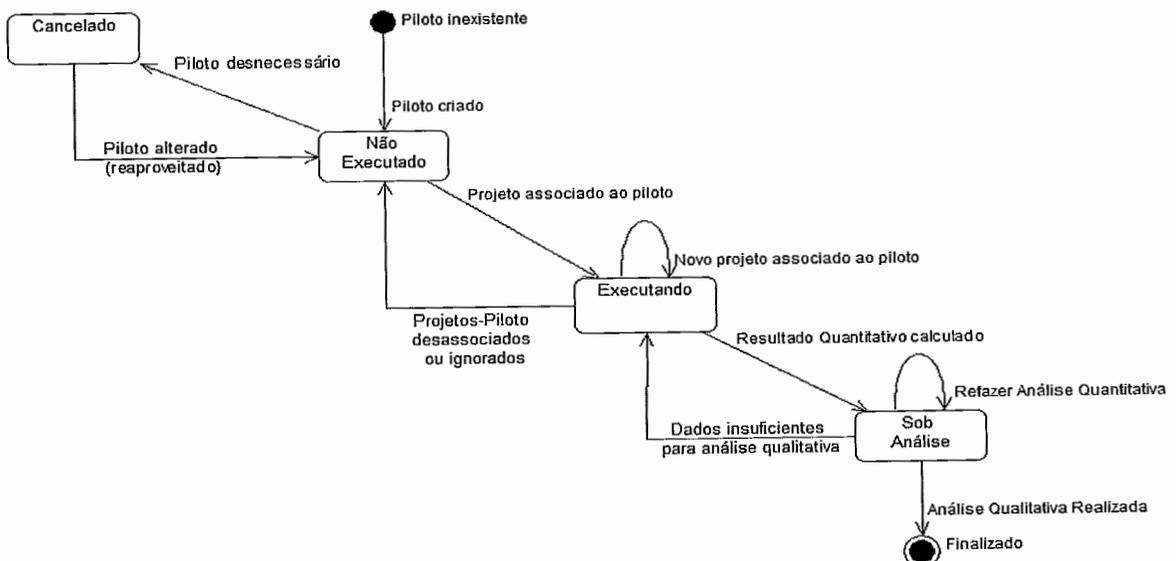


Figura 6.7 - Diagrama de estados do piloto.



A Figura 6.7 apresenta o comportamento do piloto à medida que o usuário interage com a ferramenta. A partir do momento que um piloto é criado, seu estado é Não Executado. A partir daí o piloto pode ser associado a algum projeto (esta associação implica a criação de um projeto-piloto) e então o piloto passa à situação Executando. Caso não seja necessário executar o piloto, antes de associá-lo a algum projeto, o usuário pode realizar o cancelamento do piloto. Os projetos-piloto podem ser desassociados ou ignorados enquanto o piloto encontra-se na situação Executando, caso todos os projetos-piloto sejam ignorados e/ou desassociados o piloto volta para a situação Não Executado. Quando todos os projetos-pilotos são considerados finalizados (vide estado Pronto para Análise do projeto-piloto, Figura 6.9), caso nenhum outro projeto seja associado, o usuário pode requerer que o cômputo dos resultados quantitativos, o que altera a situação do piloto para Sob Análise. O estado Sob Análise indica a realização da análise qualitativa. Neste momento, se os dados quantitativos são insuficientes para a avaliação dos efeitos do piloto, o usuário pode requerer que outros dados sejam coletados ou que outros projetos sejam associados ao piloto com o intuito de tornar os dados mais representativos. Por fim, uma vez concluída a análise qualitativa, o piloto passa para a situação Finalizado.

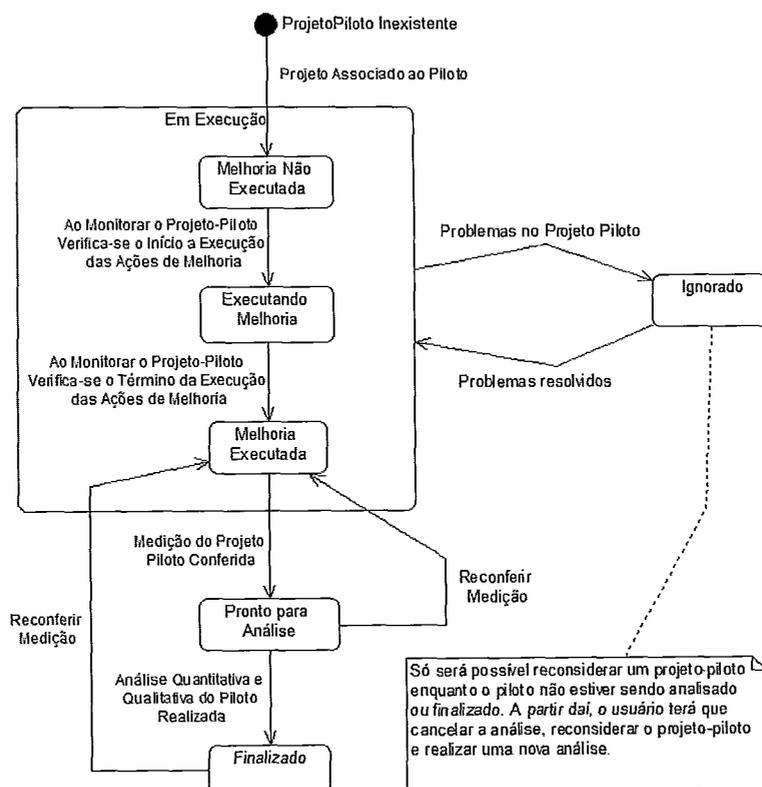


Figura 6.9 - Diagrama de estados do projeto-piloto.

No diagrama de estados do projeto-piloto (Figura 6.9), o primeiro estado assumido é Em Execução, que acontece quando o projeto é associado a um ou mais pilotos. O estado Em Execução possui três subestados: Melhoria Não Executada (a

melhoria inserida no processo ainda não foi realizada no projeto), Executando Melhoria (a melhoria está sendo executada no projeto) e Melhoria Executada (a melhoria inserida no processo já foi executada no projeto). Durante a execução, problemas no projeto-piloto (cancelamento ou suspensão do projeto) podem fazer com que o mesmo seja ignorado e não seja considerado para a análise do piloto. Caso o problema seja resolvido em tempo, o projeto poderá retornar ao estado Em Execução. Após a execução das melhorias inseridas no processo do projeto-piloto, a medição (coleta das medidas do plano de medição para o piloto) é conferida e o estado do projeto-piloto passa a ser Pronto para Análise. Em seguida, a análise quantitativa e qualitativa é realizada para o piloto (vide estado Sob Análise do piloto, Figura 6.7) e o projeto-piloto recebe o estado Finalizado.

Parte dos requisitos não funcionais está relacionada com a integração da ferramenta Pilot com a Estação TABA, inclusive no que tange a usabilidade (atendimento à interface padronizada com o usuário, conforme com as demais ferramentas que compõem a Estação TABA). Desta forma, a ferramenta foi projetada para atender a todas as restrições do ambiente e reutilizar a infra-estrutura existente, a exemplo da camada de persistência de objetos, componentes do domínio (para manipulação e utilização de processos de software e elementos relacionados) e componentes auxiliares (para manipulação de elementos gráficos, controles de interação com o usuário e outros de uso geral). A Figura 6.10 ilustra a interação direta da ferramenta Pilot com os componentes auxiliares, ao mesmo tempo em que utiliza e contribui com novos componentes do domínio.

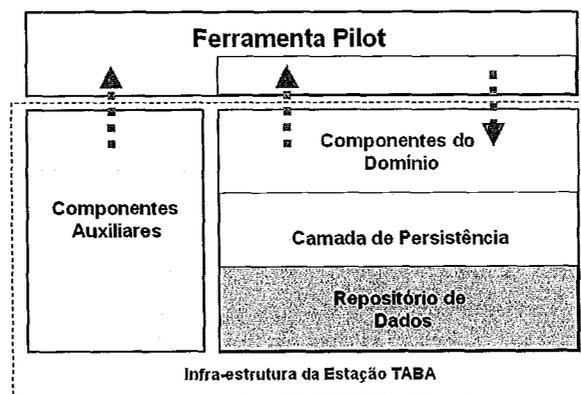


Figura 6.10 - Visão arquitetural da integração da ferramenta Pilot com a infra-estrutura provida pela Estação TABA

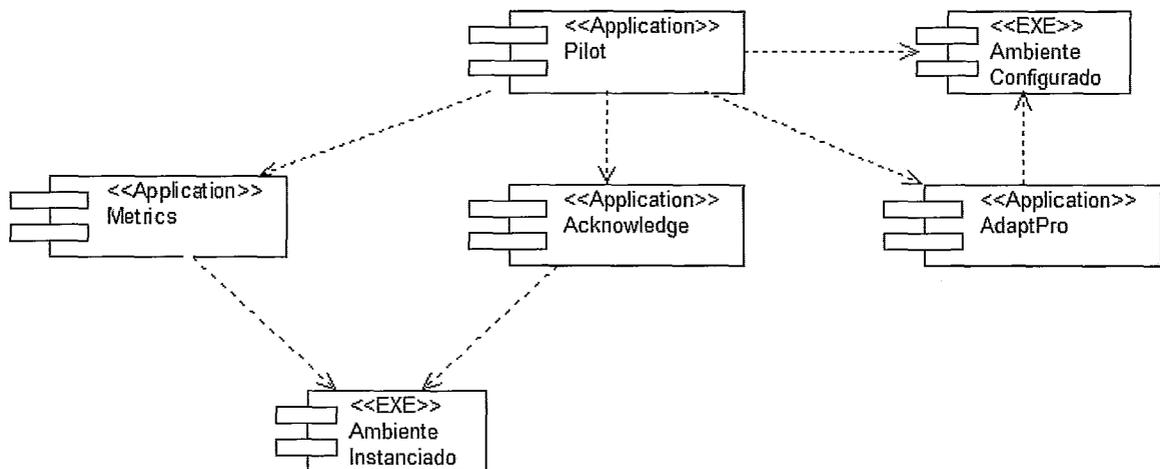


Figura 6.11 - Interação da ferramenta Pilot com aplicações que compõem a Estação TABA.

## 6.2.4. Construção

Cada ferramenta que compõe a Estação TABA contém um processo interno que é previamente cadastrado no ambiente. Este foi o primeiro passo para a implementação da ferramenta Pilot.

Em seguida, as classes foram codificadas, a interface com o usuário foi construída e as funcionalidades foram implementadas, seguindo o padrão e as diretrizes adotadas para o desenvolvimento das ferramentas integradas à Estação TABA.

### 6.2.4.1. Dificuldades encontradas

A ferramenta utilizada para a construção da Estação TABA foi o Microsoft Visual Studio (linguagem Microsoft Visual C++). Portanto, como diretriz da equipe de desenvolvimento do Projeto TABA, a utilização da referida ferramenta foi obrigatória. A inexperiência com essa ferramenta implicou em tempo e esforço para tornar-se habilitado a utilizar o ambiente e fazer uso dos recursos necessários ao desenvolvimento.

Para garantir a manutenibilidade e o funcionamento da infra-estrutura, um conjunto de restrições que regem a criação de objetos persistentes, o uso de objetos de terceiros, a definição das interfaces com o usuário dentre outras, implicaram na extensão da curva de aprendizado e do desenvolvimento das habilidades necessárias para construção da ferramenta integrada à Estação TABA.

A fase de construção ocorreu paralela à evolução da Estação TABA para permitir a interação com sistemas gerenciadores de banco de dados relacional. Durante esta evolução a camada de persistência dos objetos sofreu várias modificações e, mesmo isolada das demais camadas, um esforço adicional para

garantir o bom funcionamento da ferramenta Pilot na estrutura de armazenamento anterior e na nova estrutura teve que ser investido.

A Estação TABA é composta por mais de duas dúzias de ferramentas, possui uma estrutura complexa e uma extensa lista de classes (acima de 1500). Estas características implicam em considerável consumo de tempo para realizar compilações nas atividades de construção e testes.

#### 6.2.4.2. Facilidades

A Estação TABA foi construída sobre uma infra-estrutura de persistência de objetos que provê uma série de serviços ao desenvolvedor, principalmente no que tange o armazenamento e recuperação de objetos. Os serviços providos pela infra-estrutura contribuíram com o aumento da produtividade durante a fase de construção.

A equipe de desenvolvimento sempre esteve disponível para dirimir qualquer dúvida técnica ou conceitual relacionada à Estação TABA.

#### 6.2.5. Implantação

Após a execução dos testes, a ferramenta foi implantada nas organizações em conjunto com a atualização 2.7.0 da Estação TABA, liberada em 3 de fevereiro de 2006.

Dentre as organizações que utilizam a Estação TABA, a BL Informática, sediada em Niterói-RJ, executou no mês de março melhorias em seus processos e se propôs a utilizar a abordagem proposta nesta dissertação. A intenção da BL Informática era avaliar, através de um piloto, se as melhorias introduzidas no processo padrão da organização seriam efetivas para projetos de pequeno porte.

Neste contexto, a abordagem foi utilizada de duas formas:

(1) Para a realização de um piloto no ambiente industrial visando avaliar a utilidade e contribuição da abordagem, e

(2) Para realização de um projeto-piloto na BL Informática (capítulo 7).

### 6.3. Plano para Realização de um Piloto da Abordagem

A própria abordagem foi utilizada para apoiar um piloto da proposta descrita nesta Dissertação. O principal objetivo deste piloto é observar o esforço destinado à realização de pilotos com o uso da abordagem. Além disso, também é interesse desta avaliação inicial observar se a experiência de aplicação da abordagem contribuiu para a melhoria organizacional e se a abordagem é capaz de gerar conhecimento relevante para a organização que o utilizou.

Então, com o intuito de aproveitar a oportunidade e, ao mesmo tempo, atender a necessidade da BL Informática de realizar um piloto, um plano foi desenvolvido para condução das atividades na organização.

O plano do piloto, de acordo com a abordagem, deve conter os seguintes elementos: identificação do piloto, descrição da melhoria a ser avaliada, plano de medição, expectativas para a avaliação, critérios para seleção dos projetos que servirão de projetos-piloto e fatores que podem influenciar a avaliação. Estes elementos estão descritos a seguir.

### **6.3.1. Identificação do piloto**

Nome do Piloto: Avaliação do esforço necessário à execução de pilotos com a utilização da abordagem baseada em experimentação para a realização de pilotos.

Propósito: Quantificar o esforço necessário para executar a abordagem para avaliar a efetividade das mudanças em prol das melhorias nos processos de software.

Processo: Processo para Realização de Pilotos

Expectativa Geral: O esforço para realização do piloto não deve ultrapassar 3% do esforço total do projeto-piloto.

Sugestão do Número de Projetos para Realizar o Piloto: 1

Número de Projetos Associados: 1

### **6.3.2. Melhoria a ser avaliada**

Introdução da abordagem para avaliação da efetividade das propostas de melhorias em processos de software através da realização de projetos-piloto. A abordagem foi integrada à Estação TABA através da ferramenta Pilot, elemento que compõe a abordagem.

### **6.3.3. Plano de medição**

Objetivo 1: Quantificar o esforço necessário à avaliação da efetividade das propostas de melhoria.

Questão 1: Qual o esforço destinado à aplicação da abordagem?

Métrica 1: Esforço para o Planejamento para Realização do Piloto  
Mnemônico: EPRP                      Unidade de Medida: horas  
Procedimento Coleta: Manual, ao final do projeto.

Métrica 2: Esforço para a Monitoração do Piloto  
Mnemônico: EMP                      Unidade de Medida: horas  
Procedimento Coleta: Manual, ao final do projeto.

Métrica 3: Esforço para a Análise do Piloto  
Mnemônico: EAP                      Unidade de Medida: horas  
Procedimento Coleta: Manual, ao final do projeto.

Métrica 4: Esforço Total para a Realização do Piloto  
Mnemônico: ETRP          Unidade de Medida: horas  
Procedimento Coleta: Manual, ao final do projeto.  
Equação de Cálculo: EPRP + EMP + EAP

Questão 2: Qual o esforço total para realização dos projetos-piloto?

Métrica 5: Esforço Total dos Projetos-Piloto  
Mnemônico: ETPP          Unidade de medida: horas  
Procedimento coleta: Manual, ao final do projeto.  
Descrição: O cômputo da métrica é feito a partir da soma do esforço consumido por cada projeto-piloto concluído.  
Regras: Os projetos-piloto devem ter sido concluídos.

Questão 3: Qual o percentual do esforço consumido na aplicação da abordagem com relação ao esforço total requerido pelos projetos-piloto?

Métrica 6: Taxa de Esforço para Realização do Piloto  
Mnemônico: TERP          Unidade de Medida: %  
Procedimento Coleta: Manual, ao final do projeto.  
Equação de Cálculo: (ETRP/ETPP) \*100

Objetivo 2: Avaliar a participação na melhoria da organização

Questão 4: Qual a contribuição da abordagem para a melhoria da organização?

Métrica 7: Número de Decisões Tomadas  
Mnemônico: NDT          Unidade de Medida: unidades  
Procedimento Coleta: Manual, ao final do projeto.  
Descrição: Após a conclusão do piloto, deve ser registrado o número de decisões tomadas a partir dos resultados do piloto.

Métrica 8: Número de Lições Aprendidas  
Mnemônico: NLA          Unidade de Medida: unidades  
Procedimento Coleta: Manual, ao final do projeto.  
Descrição: No decorrer da experiência todas as lições aprendidas devem ser registradas na Estação TABA e, após a conclusão, o número de lições aprendidas deve ser informado.

Métrica 9: Número de Sugestões de Melhoria  
Mnemônico: NSM          Unidade de Medida: unidades  
Procedimento Coleta: Manual, ao final do projeto.  
Descrição: O número de sugestões de melhoria deve ser obtido a partir da análise qualitativa dos resultados do piloto.

#### 6.3.4. Expectativas para a avaliação

Taxa de Esforço para Realização do Piloto (TERP) < 3%

Número de Decisões Tomadas (NDT) > =1

Espera-se que a Taxa de Esforço para Realização do Piloto (TERP) seja inferior a 3%. Esta expectativa teve origem no estudo apresentado em Silva Filho et al. (2005), onde o esforço consumido para realização do piloto (etapa de planejamento, monitoração e análise) foi aproximadamente de 3%. Esse estudo foi realizado utilizando uma abordagem experimental e sem apoio ferramental para as atividades de planejamento, monitoração e análise dos dados.

Essencialmente, o objetivo para realizar um piloto é ter subsídios suficientes que permitam tomar decisões relacionadas à institucionalização de propostas de melhoria.

Portanto, ao menos uma decisão deverá ser tomada considerando os resultados obtidos com o piloto. A leitura que deve ser realizada quando nenhuma decisão tiver sido tomada após a conclusão do piloto é que sua realização não foi útil.

Embora não haja expectativas declaradas para as métricas Número de Lições Aprendidas e Número de Sugestões de Melhoria, estas podem indicar outras contribuições da abordagem para a melhoria organizacional.

### 6.3.5. Critérios de seleção

Considerando que o propósito deste piloto é observar o esforço requerido para utilização da abordagem e sua contribuição às organizações, a população de estudo passa a ser formada por organizações de desenvolvimento de software que aplicarão a abordagem.

Desta forma, critérios para a seleção das organizações que participaram do piloto foram definidos:

**Porte:** Pequeno ou Médio. As intenções ao definir este critério foram: minimizar a complexidade e o ruído do fator comunicação; privilegiar organizações com estruturas de decisão mais dinâmicas; realizar o piloto em uma faixa representativa da população de organizações de software;

**Nível de maturidade, capacidade ou certificação:** algum nível de maturidade ou capacidade com relação aos modelos CMMI, MPS.BR, ISO 15504 ou certificado de qualidade ISO 9001. Organizações que possuem esta característica estão mais propensas a realizar atividades em prol da melhoria. Ou, pelo menos, já possuem uma estrutura organizada a ponto de possibilitar a coleta de dados para a análise dos pilotos;

**Ambiente de Engenharia de Software:** Estação TABA, o ambiente ao qual a abordagem foi introduzida. Este critério é bastante restritivo. Entretanto, é o único ambiente de engenharia de software até o momento que incorpora a abordagem. Também seria possível omitir esta restrição e aplicar a abordagem sem o elemento ferramenta, porém, o resultado não refletiria o real potencial da abordagem;

**Estrutura organizacional:** Existência de grupo de processo ou papel equivalente. O processo que compõe a abordagem preconiza a existência deste papel dado a sua importância na condução das atividades;

**Processos institucionalizados:** É necessário que os processos estejam descritos adequadamente, especialmente o que se deseja melhorar, e que estejam sendo utilizados pela organização;

**Propostas de melhoria:** É imprescindível que a organização tenha necessidade de avaliar determinada proposta de melhoria. Sem isso não há razão para a realização de pilotos, portanto, é inútil fazer uso da abordagem.

### **6.3.6. Fatores que podem influenciar a avaliação**

#### **6.3.6.1. Curva de aprendizado**

Para aplicação plena da abordagem é necessário conhecer o funcionamento da ferramenta Pilot e ter uma visão ampla dos potenciais efeitos obtidos a partir da mudança introduzida no processo. Portanto, neste caso, que será a primeira vez que a organização irá utilizar a abordagem, o tempo destinado ao aprendizado da ferramenta poderá influenciar na quantificação do esforço para utilização da abordagem.

Para minimizar esta influência, um acompanhamento *in-loco* (*mentoring*) será realizado e as horas destinadas ao acompanhamento serão computadas. Estas horas de acompanhamento deverão ser consideradas durante a análise dos resultados do piloto.

#### **6.3.6.2. Ocorrência de retrabalho em função de falhas na ferramenta Pilot**

A ferramenta até o momento não havia sido utilizada em um ambiente industrial, conseqüentemente, embora testes tenham sido executados, há risco que falhas na ferramenta impliquem em interrupções inesperadas e perda de dados.

Para anular esta influência todo o esforço de retrabalho em função de problemas da ferramenta Pilot deverá ser registrado e, posteriormente, considerado durante a atividade de análise dos resultados.

## **6.4. Execução do Piloto da Abordagem**

Este piloto ainda não foi finalizado, sua conclusão está prevista para 7 de junho de 2006. Sendo assim, a realização do piloto será descrita até o momento da finalização desta Dissertação.

A execução do piloto teve início em 12 de abril de 2006. As diretrizes fornecidas pela abordagem para a execução do piloto foram seguidas.

## I. Atendimento aos critérios do piloto<sup>13</sup>

Todos os critérios estabelecidos para a realização do piloto foram atendidos pela organização que se propôs a aplicar a abordagem, vide Tabela 6.4.

**Tabela 6.4 - Critérios do piloto e o perfil da organização.**

	Critérios de seleção estabelecidos pelo piloto	Perfil da Organização Candidata ao Piloto da Abordagem
Porte	Pequeno ou Médio	Médio
Nível de maturidade, capacidade ou certificação	Algum nível de maturidade ou capacidade com relação aos modelos CMMI, MPS.BR, ISO 15504 ou certificado de qualidade ISO 9001	MPS.BR Nível F e ISO 9001:2000
Ambiente de Engenharia de Software	Estação TABA	Estação TABA
Estrutura organizacional	Possuir Grupo de Processo ou papel equivalente	Todos os papéis definidos no processo estão presentes na estrutura organizacional
Processos institucionalizados	Processos descritos adequadamente e que estejam sendo utilizados pela organização	Todos os processos estão descritos adequadamente, são monitorados e possuem mensuração de desempenho.
Propostas de melhoria	Necessidade de avaliar propostas de melhoria	Possui uma extensa lista de sugestões de melhoria e adota a Estratégia de Melhoria e Avaliação em Níveis (CAMPOS et al., 2005), para identificar oportunidades, priorizar ações e elaborar propostas de melhoria.

## II. Associação da organização ao piloto

Este passo formaliza a relação entre a organização e o piloto. No contexto do uso da Estação TABA esta formalização é realizada através do ambiente e acontece entre o projeto que utilizará o processo candidato à melhoria e a abordagem. Detalhes sobre esta atividade são apresentados no capítulo 7.

## III. Verificação das alterações do processo para o projeto

Este piloto trata da inclusão de uma nova tecnologia que contém um novo processo. Portanto, o passo não se aplica.

## IV. Treinamento para a equipe

O treinamento ao usuário para aplicação da abordagem foi ministrado na forma de acompanhamento (*mentoring*). Praticamente consistiu na realização de um breve ensaio para ilustrar o processo como um todo e no repasse das informações contidas no elemento guia que compõe a abordagem. Este acompanhamento consumiu 3,5 horas, das quais 2 horas foram efetivamente destinadas à execução do planejamento por parte do usuário.

<sup>13</sup>. Neste caso, os critérios devem ser aplicados à organização devido à população de estudo. Comentários a este respeito foram realizados na seção 6.3.5.

O acompanhamento também contribuiu com a identificação de vários aspectos que precisam ser melhorados no elemento guia da abordagem. Principalmente no que se refere à identificação e controle dos fatores de influência. Algumas dúvidas apresentadas pelo usuário foram de ordem conceitual, devido à fundamentação experimental da abordagem.

#### V. Planejamento e gerência dos riscos

O plano do piloto identifica os fatos que podem acontecer e as conseqüências destes fatos (riscos) no projeto.

Quando o piloto trata de projetos de desenvolvimento de software, os fatores de influência identificados no plano do piloto são acrescentados à lista de riscos do projeto e tratados como tal. O plano do piloto já sugere como o risco pode ser mitigado e/ou contingenciado.

#### VI. Inclusão das métricas no plano de medição

Neste passo, o plano de medição do projeto passa a incorporar as métricas definidas no plano de medição do piloto. Neste caso, o plano de medição do projeto é o próprio plano de medição fornecido pelo plano do piloto.

#### VII. Monitoração dos riscos e coleta das medidas

Com relação aos riscos do projeto, ações de mitigação e de contingência foram executadas.

A mitigação ocorreu com relação à influência do processo de aprendizado na avaliação do piloto. O tempo destinado ao acompanhamento foi computado isoladamente para que no momento da análise dos resultados esta influência possa ser considerada.

Medidas de contingência também foram executadas devido às ações corretivas realizadas nos dados do piloto. Atividades inerentes à evolução da Estação TABA no período que antecedeu a utilização da ferramenta Pilot na organização, especificamente no que tange ao controle de versões de processos, implicou a necessidade de realizar correções no plano do piloto. Estas correções demandaram um esforço de aproximadamente 2,5 horas (vide registro na Tabela 6.5).

Quanto à coleta das medidas, de acordo com o plano de medição, todas deveriam ser realizadas ao final do piloto. Mas, tendo em vista que o piloto será concluído apenas após a finalização da Dissertação, as medidas relacionadas ao esforço foram coletadas parcialmente para a construção de projeções.

**Tabela 6.5 - Registros de monitoração do piloto.**

Data	Situação do Piloto	Responsável	Observações
20/04/2006	Em execução	Reinaldo Cabral	O plano foi submetido à avaliação dos <i>stakeholders</i> e alterações foram sugeridas e aplicadas no plano. Além disso, foram realizadas ações corretivas para retificar os dados do plano do piloto, devido às alterações decorrentes da evolução da Estação TABA no que tange o controle de versões de processos da organização. Estas ações consumiram 3 horas, sendo 0,5h de ações de melhoria no plano e 2,5h de ações corretivas.
28/04/2006	Em execução	Reinaldo Cabral	Nenhuma ocorrência observada.
05/05/2006	---	Reinaldo Cabral	A ser realizada
12/05/2006	---	Reinaldo Cabral	A ser realizada
19/05/2006	---	Reinaldo Cabral	A ser realizada
26/05/2006	---	Reinaldo Cabral	A ser realizada
02/06/2006	---	Reinaldo Cabral	A ser realizada

### VIII. Comunicação entre os envolvidos.

A comunicação com o usuário que está realizando o piloto da abordagem tem ocorrido através de correio eletrônico, telefone e visitas *in-loco*. Até o momento a comunicação tem sido fluente e nenhum fato além dos previstos ocorreu.

## 6.5. Resultados Parciais do Piloto

Os resultados discutidos nesta seção são **parciais**. Isto é, eles denotam a situação até **02 de maio de 2006**.

### 6.5.1. Análise quantitativa parcial

Métrica	Medida
Esforço para o Planejamento para Realização do Piloto da BL Informática (EPRP)	6,5 h
Esforço para a Monitoração do Piloto da BL Informática (EMP)	0,5 h
Esforço Total para a Realização do Piloto da BL Informática (ETRP)	7 h
Esforço Total do Projeto-Piloto da BL Informática (ETPP)	80 h
Taxa de Esforço para Realização do Piloto da BL Informática (TERP)	8,75%

### 6.5.2. Fatores de influência identificados até o momento

Fator	Influenciou?	Como influenciou
Curva de aprendizado	Sim	O total de horas computadas para planejamento incluiu o tempo destinado para o acompanhamento <i>in-loco</i> , visto que orientações de como realizar cada atividade eram fornecidas antes da execução de cada passo do planejamento. O tempo líquido destinados a orientações ao usuário para realização das atividades foi de aproximadamente 1,5 h.
Ocorrência de retrabalho em função de falhas na ferramenta Pilot	Sim	Atividades inerentes à evolução da Estação TABA no período que antecedeu a utilização da ferramenta Pilot na organização, especificamente no que tange ao controle de versões de processos, implicou na necessidade de realizar correções no plano do Piloto. Estas alterações demandaram um esforço de cerca de 2,5 h.

### 6.5.3. Análise qualitativa parcial

a) As expectativas gerais para o piloto serão atendidas?

Para responder esta questão, uma projeção foi realizada considerando:

Esforço de planejamento do piloto (EPRP): 2,5 horas, sendo o esforço bruto de planejamento (6,5 h) decrementado das horas destinadas ao treinamento inicial do usuário (1,5 h) e do esforço de retrabalho em função das ações corretivas no plano do piloto (2,5h).

Esforço de monitoramento (estimado) (EMP): 2,5 horas, sendo 0,5 h já realizada e 2 h previstas para execução.

Esforço para análise dos resultados do piloto (estimado) (EAP): 2 horas.

Esforço total para realização do piloto (estimado) (ETRP): 7 horas (EPRP+EMP+EAP).

Esforço total do projeto-piloto (estimado) (ETPP): 296 horas.

Assim, temos:

$$TERP = \left( \frac{ETRP}{ETPP} \right) \times 100$$

$$TERP = \left( \frac{EPRP + EMP + EAP}{ETPP} \right) \times 100$$

$$TERP = \left( \frac{2,5 + 2,5 + 2}{296} \right) \times 100 = 2,36\%$$

Portanto, a projeção para a Taxa de Esforço para Realização do Piloto na BL Informática (TERP) seja de 2,36%, o que atende as expectativas declaradas para o piloto da abordagem.

b) Quais aspectos contribuíram para a projeção destes resultados?

O plano do piloto também foi submetido à avaliação dos *stakeholders* relevantes, que sugeriram modificações de forma a aumentar a objetividade da avaliação. Esta revisão do plano, também contribuiu com o aumento do esforço destinado ao planejamento.

c) Os resultados poderiam ser melhores? O que poderia ser feito para potencializar os resultados?

O projeto selecionado para execução do piloto da abordagem demandou um esforço muito pequeno. O uso de projetos-pilotos maiores tende a prover uma taxa menor de esforço para realização do piloto com relação ao esforço do projeto. Além

disso, o acréscimo de outros projetos associados ao mesmo piloto não implica em um aumento linear do esforço para o planejamento, monitoração e análise do piloto, visto que a etapa de planejamento, por exemplo, é executada apenas uma vez, independente do número de projetos a serem associados.

d) Que outros efeitos foram observados a partir da execução do piloto da abordagem?

Até o momento foi observado que o acompanhamento *in-loco* exerce uma influência significativa na realização das atividades, principalmente por conta do acesso rápido ao conhecimento e experiência relevante ao contexto. É provável que, apenas com o apoio do Guia para a Realização de Pilotos, o conhecimento adquirido pelo usuário que aplicou a abordagem e a qualidade dos resultados obtidos do piloto ficasse aquém daqueles obtidos nesta ocasião. Seria interessante explorar a questão.

e) Outras considerações:

Seria interessante associar outros projetos ao piloto da abordagem, com o intuito de prover mais representatividade aos dados e permitir uma observação mais acurada com relação à contribuição da abordagem na avaliação da efetividade das propostas de melhorias de software e melhoria organizacional.

## **6.6. Considerações Finais**

As estimativas para o desenvolvimento da ferramenta ficaram muito próximas do real. A previsão para conclusão do desenvolvimento foi 03 de novembro de 2005 e o encerramento se deu em 11 de novembro. Vale ressaltar que algumas pendências de documentação foram resolvidas apenas em janeiro de 2006, entre os dias 9 e 27. Neste período, com o plano de testes concluído integralmente, foram realizados novos testes e algumas operações de manutenção na ferramenta Pilot.

Durante a fase de construção as dificuldades foram restritas à administração do impacto da evolução contínua da Estação TABA com relação à integração e a inexperiência com a ferramenta MS Visual C++, que implicou em tempo e esforço para tornar-se habilitado e fazer uso dos recursos necessários ao desenvolvimento. Em contrapartida, facilidades providas pela infra-estrutura disponível e a proximidade da equipe que mantém a Estação TABA fizeram com que a construção da ferramenta Pilot ocorresse sem percalços.

A introdução da abordagem na organização foi facilitada por conta da integração com a Estação TABA. Nenhuma operação adicional teve de ser realizada, além da rotina habitual de atualização executada pelas organizações e o treinamento específico para a aplicação da abordagem. Outro benefício advindo da integração com

a Estação TABA foi a possibilidade de realizar um piloto da abordagem proposto em um contexto real na indústria.

O piloto da abordagem foi iniciado em 12 de abril de 2005 e sua conclusão está prevista para segunda semana de junho. Segundo as projeções realizadas, tudo indica que os resultados do piloto em andamento serão positivos.

Este capítulo abordou o desenvolvimento um dos elementos da abordagem proposta, a ferramenta Pilot. Também foi apresentado como a abordagem foi introduzida em uma organização de software e foi feito um breve sobre o andamento de um piloto da abordagem.

O capítulo seguinte fornece detalhes sobre o uso da ferramenta Pilot na organização BL Informática.

## **Capítulo 7. Aplicação da Abordagem na Indústria – Um Piloto na BL Informática**

---

### **7.1. Introdução**

A partir das avaliações realizadas sobre os processos da organização e das sugestões de melhoria realizadas pelos profissionais da organização, a BL Informática identificou oportunidade de melhorias e elaborou propostas para modificação dos processos.

A melhoria mais significativa inserida no processo de desenvolvimento da organização consistiu na redução do número de fases, de sete para quatro fases. Esta redução apenas tratou de reorganizar e concatenar algumas atividades, ou seja, nenhum artefato deixou de ser produzido, verificado ou avaliado, com relação à versão anterior do processo. Com relação a esta melhoria, foi observado que a redução do número de fases tornou as atividades inerentes à garantia da qualidade menos onerosas para os projetos de médio e grande porte.

Outra proposta de melhoria de alta prioridade consistia na definição de um processo de desenvolvimento de software específico para pequenos projetos, pois a versão anterior do processo (com sete fases) mostrou-se inadequada para projetos pequenos, dado o elevado percentual do custo da qualidade com relação ao custo total do projeto. Então, antes de investir esforço para a definição de um novo processo de desenvolvimento adequado a projetos pequenos, a BL Informática decidiu optar pela avaliação da efetividade da melhoria introduzida na nova versão do processo (com quatro fases) em projetos de pequeno porte.

A avaliação da efetividade da melhoria do processo de quatro fases em projetos de pequeno porte foi conduzida através da realização de um piloto. Por se tratar do primeiro piloto realizado de forma sistemática e com apoio ferramental, a BL Informática optou pela solicitação de acompanhamento pela COPPE/UFRJ para apoiar a execução das atividades.

Este capítulo relata a aplicação da abordagem para avaliação da efetividade das propostas de melhoria através de pilotos na BL Informática através da utilização do elemento ferramenta Pilot.

## 7.2. A BL Informática

A BL Informática é uma empresa cujo principal foco de negócio é o desenvolvimento, manutenção, integração e fábrica de software. Sediada em Niterói-RJ, tem seu núcleo composto por mais de 30 profissionais. Criada em 1987, a organização vem acumulando conhecimento nas diversas áreas em que atua e tem investido continuamente em treinamento em prol da produtividade e do incremento da qualidade de seus produtos.

Em 2003, a BL Informática empreendeu na implantação de processos visando obter a certificação ISO 9001:2000 (ISO, 2000), alcançada em dezembro de 2004. Após isso, a empresa traçou uma estratégia visando atingir o nível de maturidade 3 do CMMI em 2006. Parte desta estratégia incluía a obtenção do nível F do MPS.BR, o qual foi alcançado em setembro de 2005.

No momento, a BL Informática está em vias da avaliação SCAMPI para obtenção do nível 3 de maturidade do CMMI. Na pré-avaliação (*readiness*), a empresa alcançou a marca expressiva de 98,2% de evidências adequadas encontradas.

Deste ponto em diante faremos referência a BL Informática apenas como BL.

## 7.3. Planejamento do Piloto na BL Informática

Nas seções subseqüentes será apresentado o uso da ferramenta Pilot pela BL no planejamento do piloto para avaliar a efetividade do processo de quatro fases em projetos pequenos com relação à diminuição do custo da qualidade.

### 7.3.1. Identificação do piloto

O acesso à ferramenta Pilot foi realizado através do Ambiente Configurado da Estação TABA<sup>14</sup> para a BL. A primeira atividade disponível é destinada à identificação do piloto (Figura 7.1).

O primeiro passo do planejamento consistiu em informar os dados essenciais para realização do piloto. A identificação envolveu a escolha de um nome para o piloto, a declaração do tipo do processo candidato à melhoria, o número de projetos que deverão ser associados ao piloto (projetos-piloto), o processo candidato à melhoria, o propósito e a expectativa geral para a realização do piloto.

O nome dado ao piloto pela BL Informática foi 'Avaliação da Utilização do Processo de 4 Fases em Projetos Pequenos', nome pertinente ao problema investigado e que sucintamente descreve a razão da execução do piloto.

---

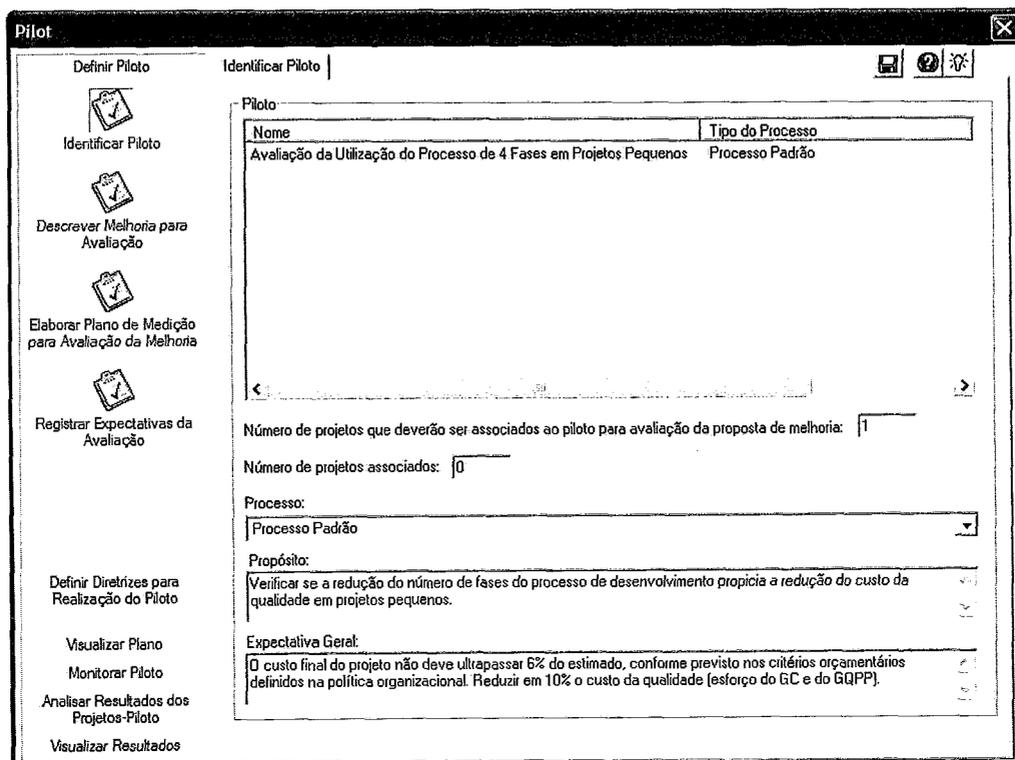
<sup>14</sup> Ambiente onde são elaborados os planos para os processos dos projetos. O acesso à maioria das ferramentas que atuam no nível organizacional é feito a partir desse ambiente.

O tipo do processo selecionado foi “Processo Padrão”, pois a alteração do número de fases ocorreu no processo padrão da organização, ou seja, todos os demais processos especializados ou instanciados também refletiram esta mudança. A ferramenta Pilot fornece a lista de processos de acordo com o tipo processo escolhido.

A primeira decisão que foi tomada durante o uso da ferramenta Pilot foi a respeito do número de projetos que deveriam ser associados ao piloto. A BL optou por iniciar a experiência com apenas um projeto. Caso ao final do piloto, a análise qualitativa sugerisse a execução de outros projetos-piloto para melhor subsidiar a tomada de decisão, a BL realizaria a alteração do número de projetos a serem associados e executaria outros projetos-piloto.

É importante ressaltar que quanto maior for o número de projetos-piloto, mais representativo será o resultado obtido. No entanto, fatores como custo e a disponibilidade de tempo para a tomada de decisão devem ser considerados no momento de decidir a respeito deste número.

Ainda na identificação do piloto, a BL evidenciou a expectativa geral para a experiência: “O custo final do projeto não deve ultrapassar 6% do estimado [...]” e “Reduzir em 10% o custo da qualidade [...]”. Estas informações são fundamentais para que seja possível avaliar se as mudanças surtiram o efeito esperado para o piloto. O plano de medição e os fatores de influência são elaborados a partir destas informações.



Nome	Tipo do Processo
Avaliação da Utilização do Processo de 4 Fases em Projetos Pequenos	Processo Padrão

Número de projetos que deverão ser associados ao piloto para avaliação da proposta de melhoria: 1

Número de projetos associados: 0

Processo: Processo Padrão

Propósito: Verificar se a redução do número de fases do processo de desenvolvimento propicia a redução do custo da qualidade em projetos pequenos.

Expectativa Geral: O custo final do projeto não deve ultrapassar 6% do estimado, conforme previsto nos critérios orçamentários definidos na política organizacional. Reduzir em 10% o custo da qualidade (esforço do GC e do GQPP).

Figura 7.1 - Tela Identificar Piloto.

### 7.3.2. Descrição da melhoria a ser avaliada

Neste passo as ações de melhoria contidas na proposta são descritas. Cada ação de melhoria pode consistir na alteração de atividades (inclusão, alteração, exclusão ou reordenação), utilização de uma nova ferramenta de apoio, utilização de novos métodos e/ou técnicas na execução de determinadas atividades, alteração ou inserção de novos modelos e diretrizes.

As ações de melhoria para avaliação da BL não estavam associadas a uma atividade específica do processo, visto que implicavam a redução do número de fases do processo e na reorganização das atividades. Por isso, apenas uma descrição textual foi realizada. Caso alguma atividade do processo tivesse sido alterada, seria necessário descrever a mudança através da interação com o processo apresentado na forma de árvore (Figura 7.2). Esta interação consiste na escolha da atividade e, em seguida, no tipo de operação desejada (alteração, exclusão, reordenação ou criação de uma nova atividade).

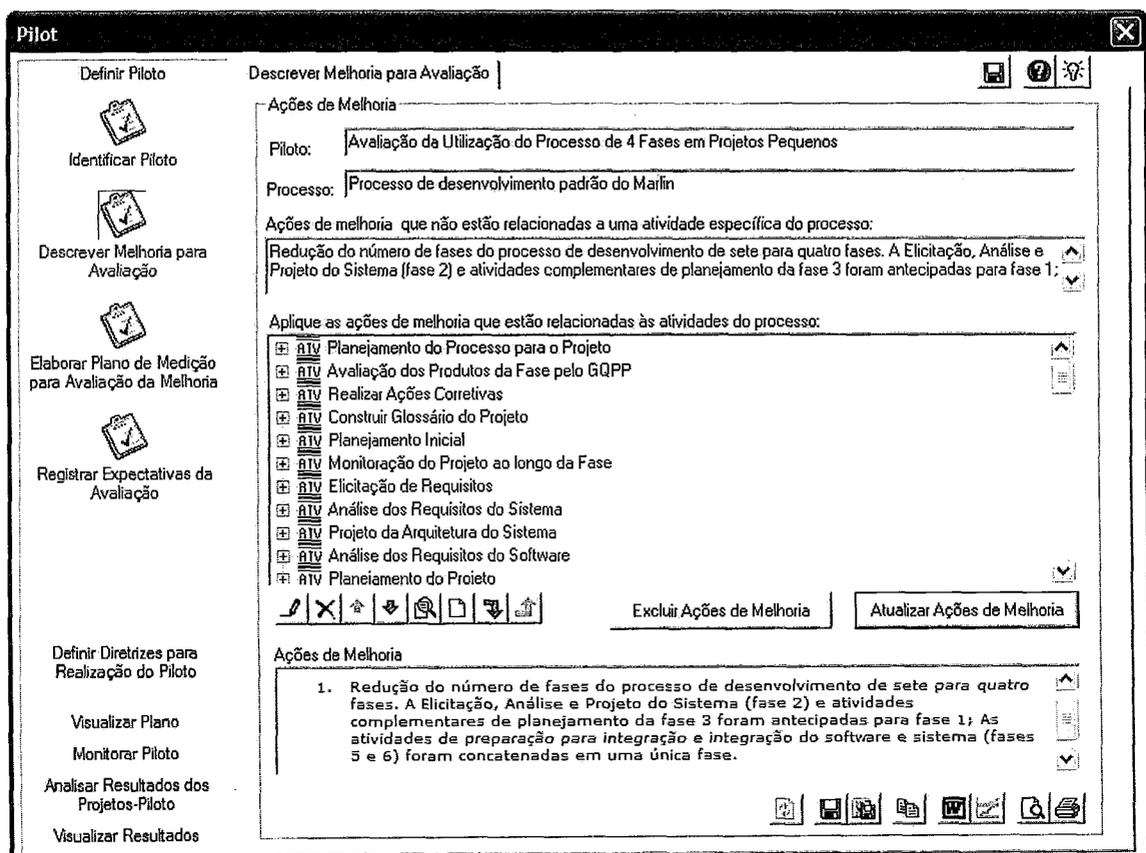


Figura 7.2 - Tela Descrever Melhorias para Avaliação.

### 7.3.3. Elaboração do plano de medição

Com base nas expectativas declaradas para o piloto, a BL elaborou um plano de medição de forma que fosse possível observar os efeitos do processo de quatro fases em projetos pequenos. De acordo com a expectativa geral, o efeito esperado com o

uso do processo estava relacionado com a diminuição do custo da qualidade. Sendo assim, aplicando o método GQM (VAN SOLINGEN; BERGHOUT, 1999), a organização definiu objetivos, questões e métricas para compor o plano de medição do piloto.

Além do objetivo “Diminuir o custo da qualidade”, a BL identificou outros objetivos que poderiam auxiliar a análise dos resultados: “Monitorar o processo Monitoração e Controle do Projeto” e “Monitorar o processo de Gerência de Requisitos”. A monitoração e controle do projeto estão relacionados com medidas de índice de desempenho e valor agregado. A monitoração da gerência dos requisitos visa observar o índice de alteração nos requisitos do projeto. Embora estes objetivos não estejam diretamente relacionados ao propósito do piloto, eles podem ilustrar variações que afetam os dados coletados para o objetivo principal. Por exemplo, caso os requisitos apresentem instabilidade elevada, o custo da qualidade será incrementado por conta do número de avaliações requeridas para as novas versões dos artefatos. Caso este evento ocorra, o resultado quantitativo irá demonstrar que o processo não foi atendeu às expectativas declaradas. Isto pode não ser verdade.

**Pilot**

Definir Piloto | Elaborar Plano de Medição

Identificar Piloto

Descrever Melhoria para Avaliação

Elaborar Plano de Medição para Avaliação da Melhoria

Registrar Expectativas da Avaliação

Definir Diretrizes para Realização do Piloto

Visualizar Plano

Monitorar Piloto

Analisar Resultados dos Projetos-Piloto

Visualizar Resultados

**Objetivos:**

Nome	Descrição
<input checked="" type="checkbox"/> Diminuir o custo da qualidade	
<input checked="" type="checkbox"/> Monitorar o processo Gerência de Requisitos	
<input type="checkbox"/> Melhorar estimativa de tempo	
<input type="checkbox"/> Melhorar estimativas de custo	

**Questões:**

Nome	Descrição
<input checked="" type="checkbox"/> Qual o custo da qualidade (esforço) do GC?	
<input checked="" type="checkbox"/> Qual o custo da qualidade (esforço) do GQPP?	Custo da qualidade do GQPP co...

**Métricas:**

Nome	Descrição
<input checked="" type="checkbox"/> Esforço do GC	Esforço realizado pelo GC
<input checked="" type="checkbox"/> Percentual de Esforço do GC	Custo da qualidade do GC com r...

Geral | Composição da métrica | Procedimento de análise | Procedimento de coleta

Mnemônico: PEGC      Atomicidade: Não atômica

Equação cálculo: EGC/ERP      Unidade de medida: %

Valor base: 0.00

Figura 7.3 - Tela Elaborar Plano de Medição.

### 7.3.4. Registro das expectativas para o piloto

Uma vez definida as métricas para possibilitar a mensuração dos efeitos de interesse, a BL declarou os resultados esperados para cada métrica diretamente relacionada com a expectativa geral (Figura 7.4). A declaração destas expectativas possibilita a execução de uma análise quantitativa automática com base nas medições realizadas. A ferramenta Pilot compila as medidas coletadas e provê uma análise quantitativa informando se o piloto atendeu, não atendeu ou atendeu parcialmente às expectativas.

Vale ressaltar que é importante justificar a fonte das expectativas. Em outras palavras, é preciso demonstrar que a expectativa é atingível e é coerente com o propósito do piloto. Esta preocupação facilita a obtenção do comprometimento dos envolvidos.

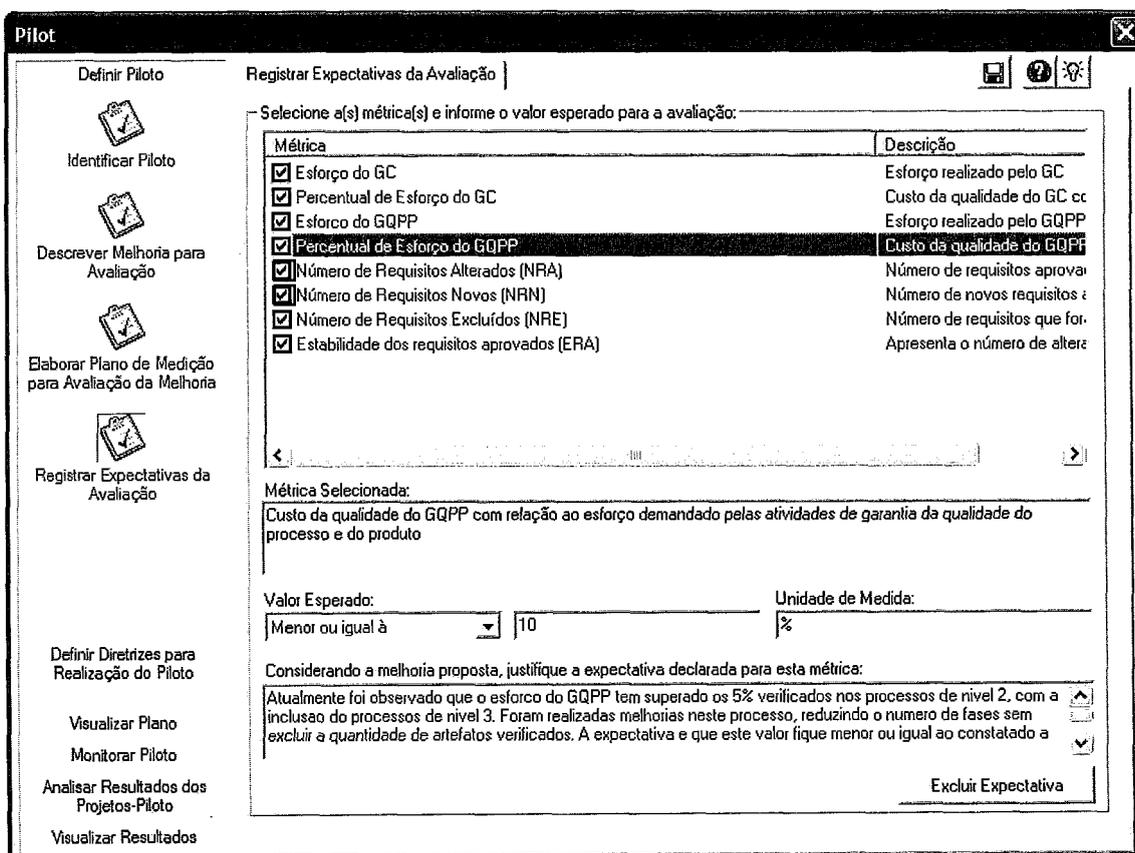


Figura 7.4 - Tela Registrar Expectativas da Avaliação.

### 7.3.5. Critérios para seleção de projetos

O propósito declarado para o piloto requer que a execução do piloto seja realizada em projetos pequenos. Além disso, para permitir uma avaliação com alguma representatividade, a BL estabeleceu critérios que permitissem a seleção de projetos que representassem o caso médio com relação aos projetos pequenos desenvolvidos pela organização.

A tela da ferramenta Pilot para esta atividade (Figura 7.5) também permite que outros critérios sejam informados através do uso do campo textual 'Outras Recomendações'.

Estes critérios funcionam como recomendação para a seleção do projeto que será associado ao piloto (projeto-piloto). Recomenda-se que, caso haja alguma divergência, a decisão de associar o projeto ao piloto seja tomada em conjunto pelos envolvidos.

The screenshot shows the 'Pilot' application window with the title 'Estabelecer Critérios para Seleção do Projeto'. The interface is divided into a sidebar on the left and a main content area on the right. The sidebar contains the following options: 'Definir Piloto', 'Definir Diretrizes para Realização do Piloto', 'Estabelecer Critérios para Seleção do Projeto' (highlighted), 'Identificar Fatores de Influência', 'Visualizar Plano', 'Monitorar Piloto', 'Analisar Resultados dos Projetos-Piloto', and 'Visualizar Resultados'. The main content area is titled 'Estabelecer Critérios para Seleção do Projeto' and contains several sections: 'Projeto' with a 'Paradigmas' list (Estruturado, Orientado a Objetos) and a 'Tipos de Software' list; 'Modelo de Ciclo de Vida' with a 'Nome' field; 'Tamanho da Equipe' with a range selector (Entre [ ] e [ ] membros); 'Complexidade dos Projetos' with radio buttons for Baixa, Média, and Alta; 'Nível de Experiência da Gerência' with radio buttons for Baixo, Médio, and Alto; 'Tamanho dos Projetos' with radio buttons for Pequeno (<14 homem-mês), Médio (<96,5 homem-mês), and Grande (>= 96,5 homem-mês); 'Duração Prevista' with a range selector (Entre [1] e [3] meses); 'Experiência da Equipe com a Tecnologia' with radio buttons for Baixa, Média, and Alta; 'Experiência da Equipe em Engenharia de Software' with radio buttons for Baixa, Média, and Alta; and 'Outras Recomendações' with a text input field.

Figura 7.5 - Tela Estabelecer Critérios para Seleção do Projeto.

### 7.3.6. Identificação dos fatores de influência

A identificação dos fatores que podem exercer alguma influência nos resultados do piloto é realizada com base na lista de fatos que podem ocorrer durante a execução do projeto. Cada fator de influência possui atributos que indicam como o fator pode influenciar os resultados, como identificar a influência do fator e o que pode ser feito para minimizar ou anular o efeito causado pelo fator no resultado do piloto.

Para a BL um fato relevante que pode ocorrer e influenciar a análise do piloto é se, em algum momento do projeto, a equipe que participa do desenvolvimento ficar indisponível, principalmente aqueles que assumem o papel do GQPP ou GC. Se algumas dessas pessoas, que possuem experiência no desempenho das atividades

de avaliação, forem substituídas, o esforço (em horas) para realizar a avaliação dos artefatos tende a aumentar, como conseqüência o custo da qualidade também tenderá a aumentar, o que afetaria os resultados do piloto. Para tratar o fator de influência, caso o fato ocorra, um profissional externo, com as mesmas habilidades, será contratado para realizar as tarefas afetadas. A Figura 7.6 ilustra o registro desse fator de influência.

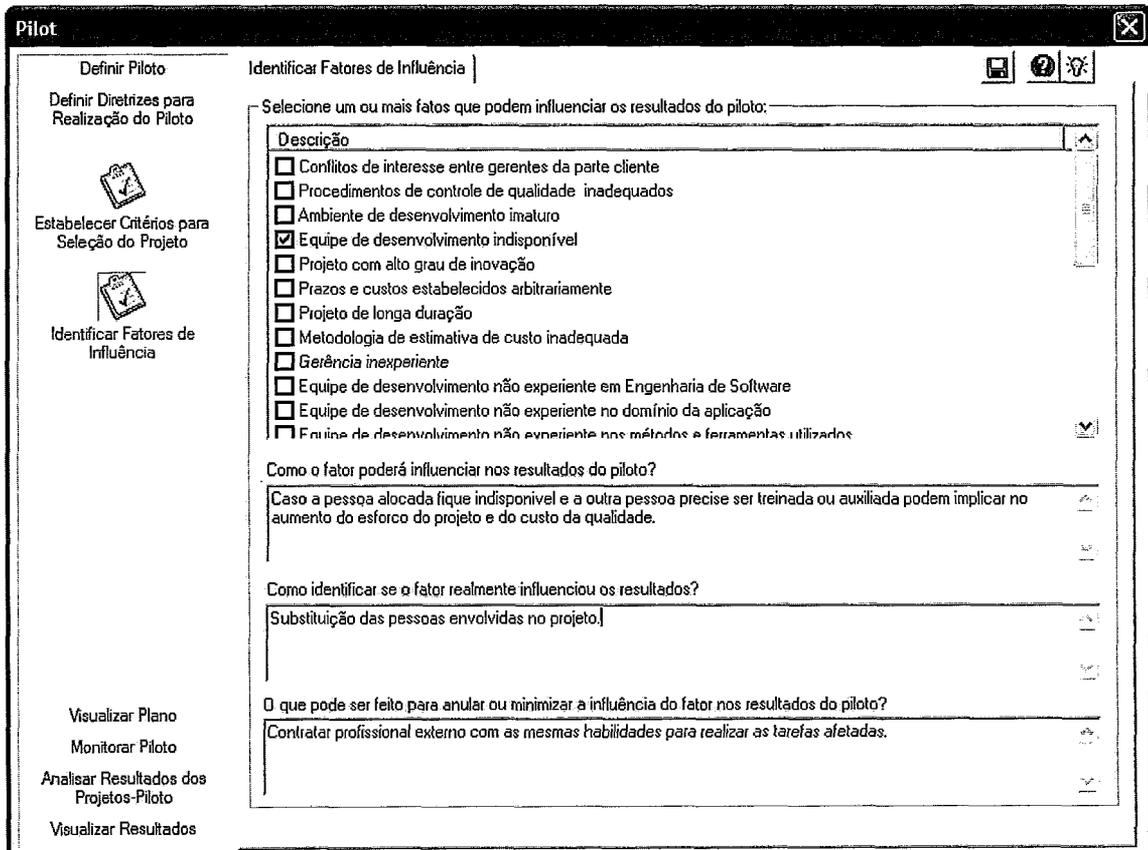


Figura 7.6 - Tela Identificar Fatores de Influência.

### 7.3.7. Emissão do plano do piloto e do documento de diretrizes

Após passar por cada atividade inerente ao planejamento, a BL pode observar e acessar sempre que desejar o plano para realização do piloto (Figura 7.7). Além deste plano, a BL encaminhou o documento 'Diretrizes para Realização do Piloto' para o gerente do projeto. Este documento contém um conjunto de passos que visam orientar o gerente do projeto a conduzir o piloto atendendo a todas as recomendações estabelecidas no plano do piloto.

O plano do piloto e o documento contendo as diretrizes para realização do piloto na BL Informática, constam integralmente nos Anexos 2 e 3 desta Dissertação.

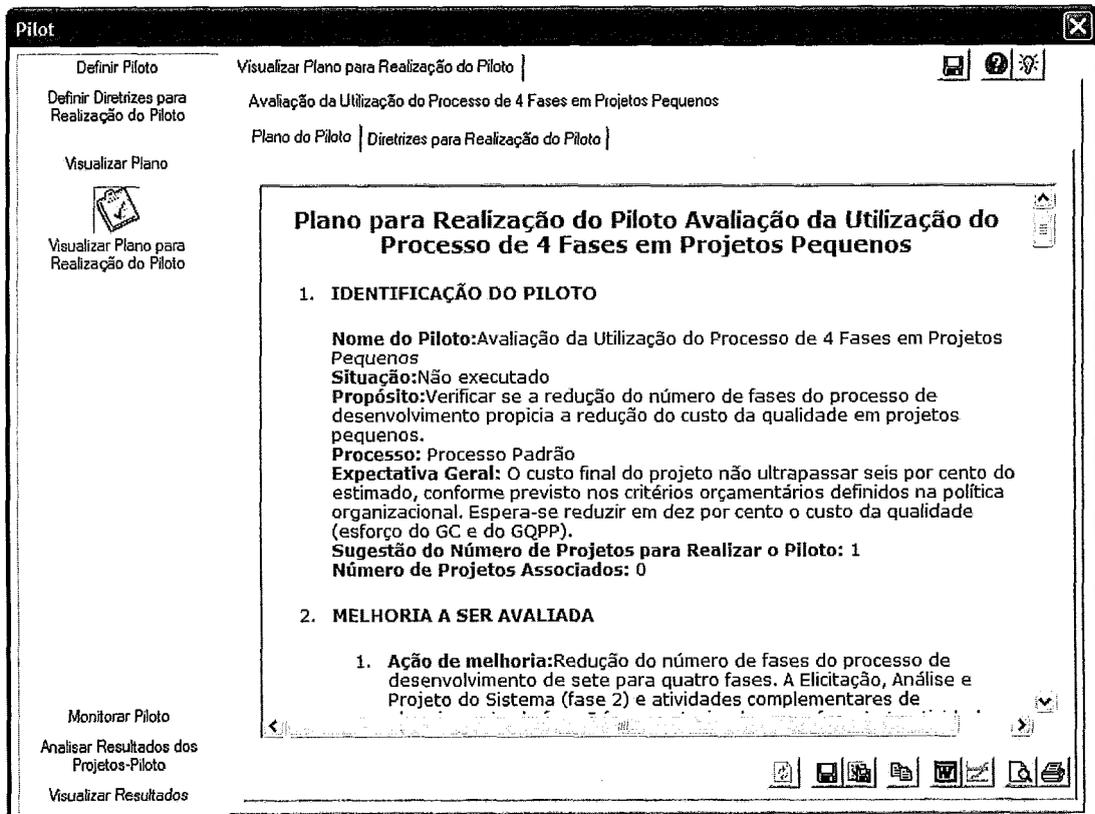


Figura 7.7 - Plano para Realização do Piloto.

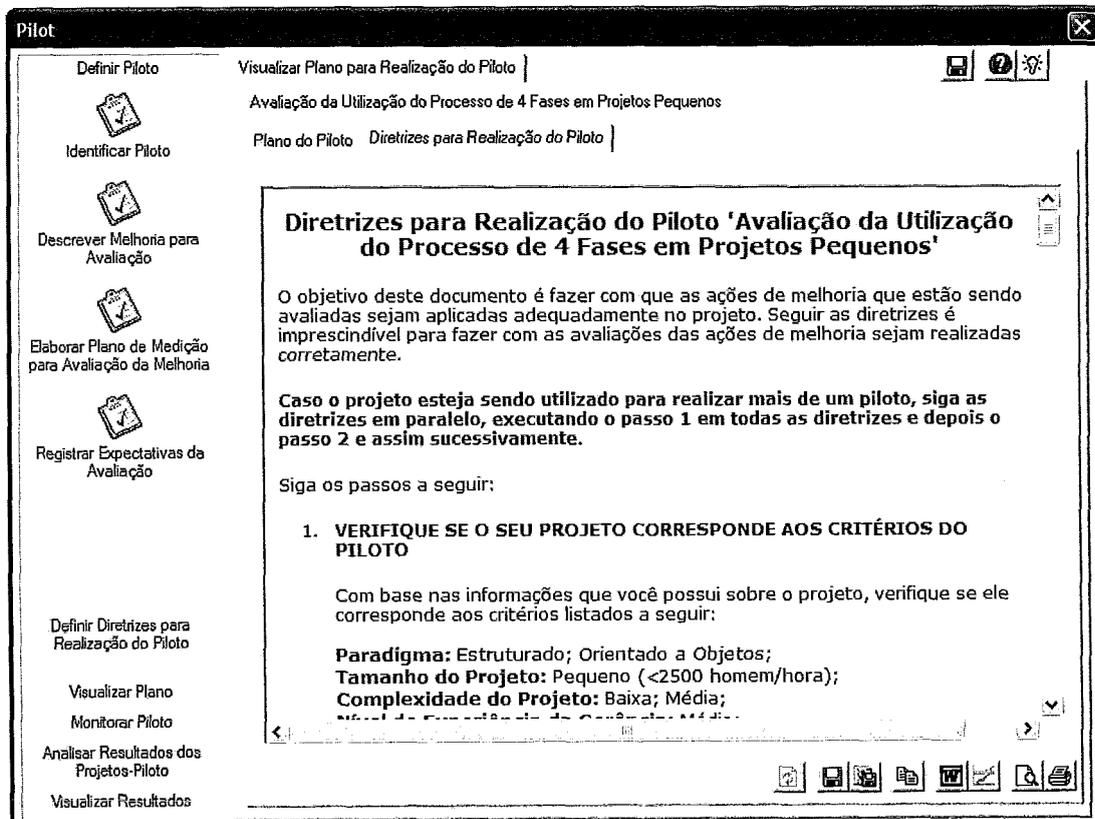


Figura 7.8 - Diretrizes para o gerente do projeto.

## **7.4. Execução do Piloto na BL Informática**

Concluído o plano e distribuídas as diretrizes para os gerentes de projeto, é iniciada a execução do piloto. Primeiro, o projeto é associado ao piloto. Em seguida, a agenda de monitoração é definida e o gerente do projeto prossegue o desenvolvimento observando as diretrizes para a realização do piloto.

### **7.4.1. Associação do projeto ao piloto**

Um dos gerentes de projeto da BL, antes de planejar o processo para o projeto, recebeu as diretrizes para realização do piloto e seguiu os passos descritos nas diretrizes.

O primeiro item observado foi se as características do projeto atendem aos critérios estabelecidos pelo piloto. Como os critérios foram atendidos, o projeto BL 00177/2006 foi associado ao piloto caracterizando-o como projeto-piloto. Esta associação foi realizada através da ferramenta AdaptPro<sup>15</sup>.

O projeto BL 00177/2006 visa realizar a migração de um sistema na plataforma Linux para plataforma Windows. O início do projeto estava previsto para 17 de abril de 2006 e seu encerramento para 7 de junho de 2006.

Após a associação do projeto ao piloto, o plano do processo para o projeto-piloto foi avaliado para verificar se o plano do processo estava de acordo com o estabelecido pelas diretrizes para a realização do piloto. A Figura 7.9 ilustra a associação entre o projeto e o piloto na tela da ferramenta AdaptPro do Ambiente Instanciado da Estação TABA para o Projeto BL 00177/2006.

### **7.4.2. Monitoração do piloto**

Em seguida, a agenda de monitoração do piloto foi definida pelo grupo de processo e informada ao gerente e demais envolvidos com a experiência (GQPP e GC).

Até o momento foram realizados dois registros de monitoração para o piloto, num total de cinco previstos. O documento utilizado pela BL para monitoração consta no Anexo 4 desta Dissertação.

---

<sup>15</sup> A AdaptPro (BERGER, 2003) é uma ferramenta que integra a Estação TABA e apóia a caracterização dos projetos, definição dos ciclos de vida e elaboração do plano do processo para os projetos, além de prover um ambiente instanciado para execução de cada projeto. Estas funcionalidades estão disponíveis na versão da ferramenta AdaptPro que é executada no Ambiente Configurado da Estação TABA. Uma versão simplificada da ferramenta, que permite consultar apenas os dados do processo instanciado, é executada no Ambiente Instanciado da Estação TABA (ambientes de cada projeto).

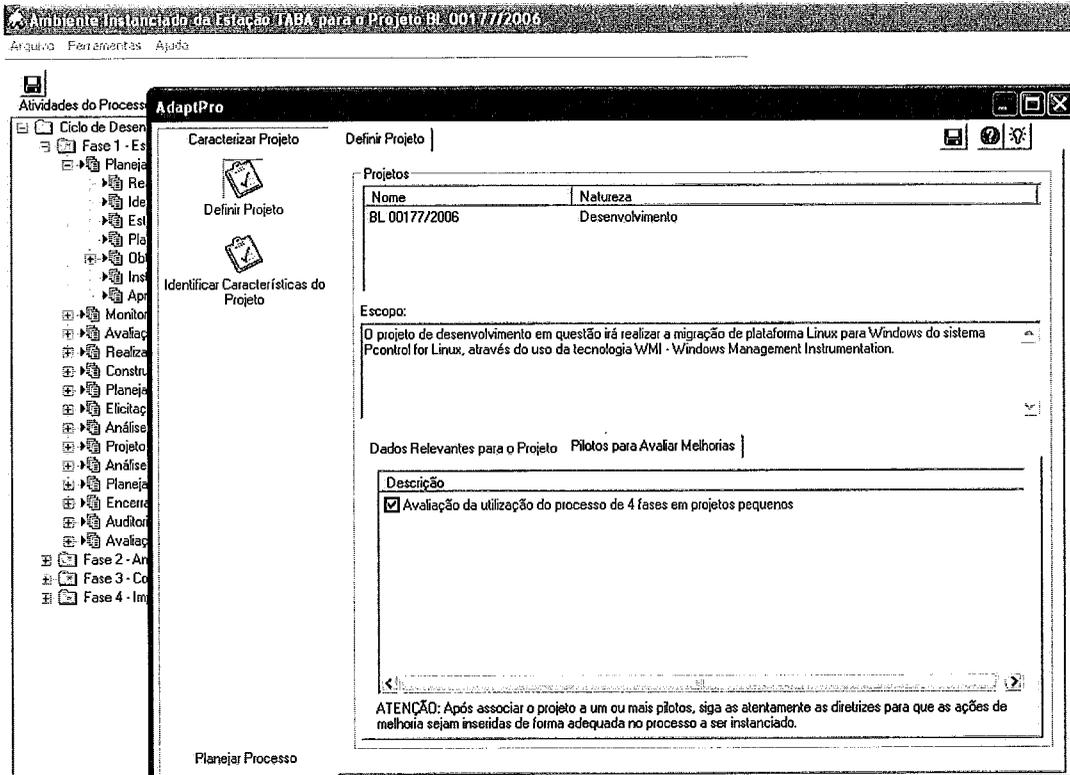


Figura 7.9 - Projeto BL 00177/2006 instanciado na Estação TABA e associado ao piloto.

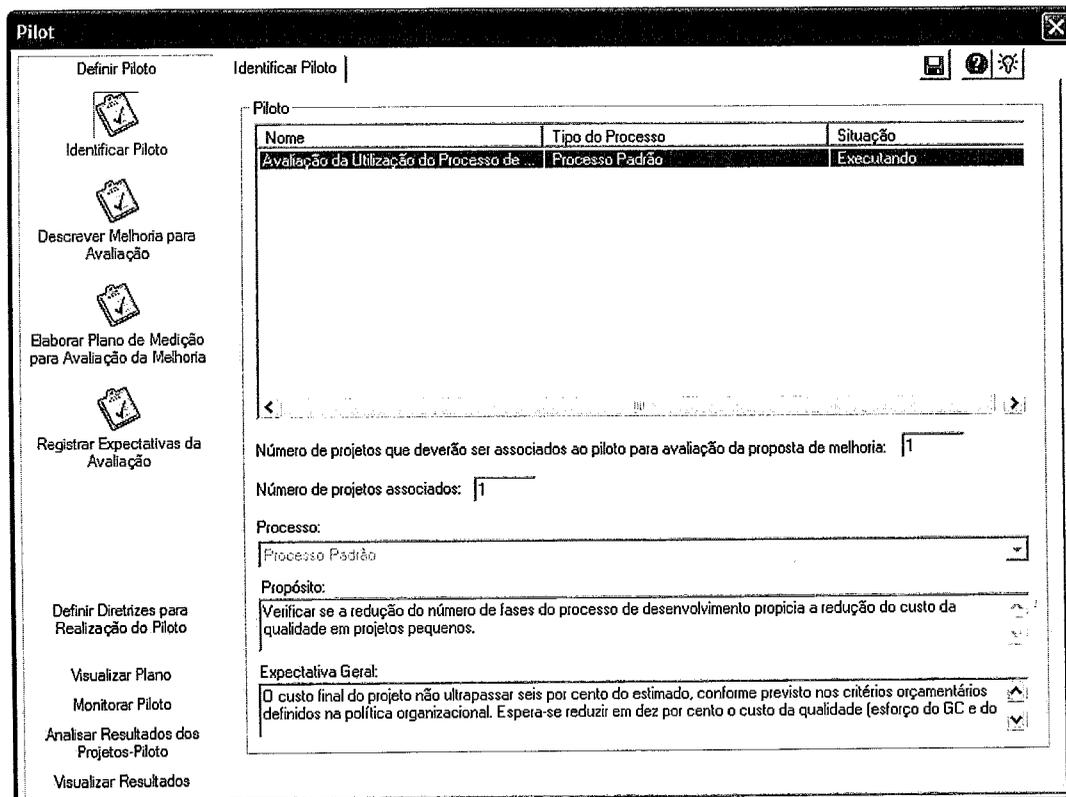


Figura 7.10 - Tela Identificar Piloto após a associação do projeto BL 00177/2006 ao piloto.

### **7.4.3. Execução do projeto-piloto**

O gerente do projeto, seguindo as diretrizes, incluiu no plano de medição do projeto as métricas destinadas à avaliação do piloto. Este procedimento auxilia a condução das atividades de duas formas: facilita o trabalho da equipe do projeto no que tange à coleta das medidas, pois todos os registros podem ser realizados em apenas uma ferramenta (Metrics<sup>16</sup>); e fornece as condições necessárias para que a ferramenta Pilot possa obter, automaticamente, as medições relevantes para o piloto a partir dos dados coletados.

Seguindo ao mesmo princípio, o gerente também tratou o fator de influência como risco associados ao projeto. Assim, ele incluiu o fator de influência na lista de riscos do projeto e desenvolveu um plano que permitisse monitorar e realizar ações de mitigação e contingência para cada risco de forma centralizada, através da ferramenta RiscPlan<sup>17</sup>.

### **7.5. Considerações Finais**

Os dados coletados até a presente data (12 de maio) ainda não oferecem indicativo dos resultados do piloto na BL Informática. Por enquanto, os requisitos estão estáveis e o projeto tem sido executado conforme o previsto.

Os dados expressos nos registros de monitoração coletados junto aos envolvidos no piloto sugerem que a mudança facilitou gerenciamento do tempo. Porém esta informação, aparentemente, não está relacionada com o propósito do piloto.

---

<sup>16</sup> A Metrics é uma ferramenta que integra a Estação TABA e provê mecanismos de coleta automática de medidas. A ferramenta também dispõe de uma interface para que o usuário também possa realizar o registro manual das medidas.

<sup>17</sup> Ferramenta RiscPlan apóia o planejamento e a gerência de riscos do projeto (FARIAS, 2002).

## Capítulo 8. Considerações Finais

---

### 8.1. Conclusão

As organizações precisam continuamente se ajustar às intempéries do mercado para se manterem competitivas. Investimentos têm sido feitos em busca do aumento da eficiência, diminuição do custo e melhoria da qualidade dos produtos na perspectiva do cliente. Neste contexto, a melhoria contínua dos processos da organização é um elemento chave. Modificar os processos e adaptá-los às novas necessidades não é tarefa trivial. Uma proposta de mudança que implique em uma alteração mal sucedida nos processos organizacionais pode levar a organização a uma situação turbulenta e perigosa.

Com intenção de balizar as iniciativas de melhoria, modelos e normas em prol da qualidade sugerem várias práticas que podem ser utilizadas para avaliar a efetividade das propostas de melhoria antes que estas sejam institucionalizadas nos processos organizacionais. Porém, estes modelos não descrevem como estas práticas devem ser conduzidas para prover avaliações com razoável nível de acurácia.

Estudos experimentais, simulação, *benchmarking* e o uso de pilotos foram apresentados nesta Dissertação como mecanismos que podem auxiliar na avaliação das propostas de melhoria.

Considerando a sugestão dos modelos de qualidade de processos de software em adotar pilotos para avaliar as propostas de melhoria, um estudo baseado em revisão sistemática foi realizado com o intuito de identificar instrumentos que facilitassem a realização de pilotos no ambiente industrial. O estudo foi útil na identificação de alguns instrumentos para facilitar o uso de pilotos. No entanto, os instrumentos identificados eram aplicados a contextos específicos, não haviam sido avaliados no âmbito industrial ou não atendiam a todas as práticas sugeridas por modelos em prol da qualidade.

Visando suprir esta lacuna e apoiar as organizações a institucionalizar melhorias efetivas em seus processos de software, foi descrita uma abordagem para realização de pilotos, de forma sistemática, planejada e controlada, que permitisse avaliar os efeitos das propostas de mudança nos processos candidatos à melhoria.

A abordagem é composta por três elementos:

- Um processo, que consiste em um conjunto ordenado de atividades, com os respectivos insumos, papéis e critérios de entrada e saída, além de modelos para a produção de artefatos gerados;
- Uma ferramenta, responsável pela interação entre o usuário, a abordagem e o ambiente de engenharia de software centrado em processo. Atua como um facilitador propiciando a automação de algumas atividades do processo;
- Um guia, que provê conhecimento e diretrizes para apoiar a realização das atividades do processo. O conhecimento fornecido pelo elemento guia pode ser consultado através do elemento ferramenta ou diretamente no artefato que contém o guia.

Para potencializar a aplicação da abordagem, de forma a tirar proveito das facilidades providas pelos ambientes de engenharia de software centrados em processos, a abordagem foi introduzida na Estação Taba através da integração do ambiente com um dos elementos da abordagem, a ferramenta Pilot.

Com o intuito de observar a aplicação da abordagem em um ambiente real e analisar a sua utilidade, um piloto para aplicação da própria abordagem foi planejado e iniciado. A aplicação da abordagem em um projeto real na indústria foi planejada e iniciada pela empresa BL Informática, que necessitava avaliar a efetividade da melhoria da nova versão do seu processo de desenvolvimento para projetos pequenos. Embora ambas as experiências ainda não tenham sido concluídas, projeções indicam que a abordagem é útil e que poderá contribuir efetivamente com as iniciativas de melhoria nas organizações.

É importante ressaltar que a introdução da abordagem na BL Informática foi facilitada devido à integração da ferramenta Pilot com a Estação TABA. Esta integração viabilizou a realização do piloto da abordagem em um contexto real na indústria.

O produto desta Dissertação também pode ser considerado uma contribuição para a Estação TABA, visto que ele agrega ao ambiente um mecanismo útil para apoiar as avaliações das propostas de melhoria de processos de software, tal como a Estratégia de Avaliação e Melhoria em Níveis adotada pela Estação TABA prescreve.

O uso da abordagem proposta também auxilia as organizações que adotam e/ou têm interesse em obter níveis superiores em avaliações com relação aos modelos de qualidade CMMI e MPS.BR, pois a abordagem cobre todas as práticas e fornece os

indicadores inerentes à avaliação de propostas de melhorias e realização de pilotos sugeridos por esses modelos.

A abordagem também facilita a construção de conhecimento organizacional de forma incremental. Cada piloto realizado possibilita o incremento da base de lições aprendidas e propicia o auto-conhecimento organizacional. Estes benefícios podem ser potencializados através da replicação das experiências de forma sistemática, que viabilizam a agregação de dados de uma mesma família de experiências e permitem a construção de evidências aplicáveis ao contexto da organização<sup>18</sup>.

As experiências que foram iniciadas para avaliar a utilidade da abordagem serão concluídas. Novas experiências serão realizadas na indústria com o intuito de obter maior visibilidade dos efeitos da abordagem e colher subsídios para realizar refinamentos, caso necessário.

## **8.2. Perspectivas Futuras**

Como mencionado no capítulo 2, a simulação de processos de software é uma alternativa ao uso de pilotos, especialmente quando há restrições de tempo para decidir sobre a inserção da melhoria no processo ou quando o risco da proposta de melhoria é alto a ponto de inviabilizar o seu teste em um projeto real.

Uma das questões que se apresenta como grande óbice à ampla adoção de modelos dinâmicos para simulação nas organizações, é a dificuldade para estabelecer as relações entre as variáveis do mundo real que influenciam os processos, além da necessidade ajustar os modelos ao perfil da organização. Uma vez que estas relações possam ser capturadas e quantificadas de forma eficiente, é possível que o uso de simulação dinâmica de processos torne-se um instrumento acessível às pequenas e médias empresas.

A aplicação da abordagem para realização de pilotos no âmbito organizacional poderá auxiliar a construção de modelos dinâmicos acurados para simulação de processos. Cada piloto realizado pode permitir que a organização estabeleça, quantitativamente, relações entre fatores que influenciam os projetos.

Estudos com o intuito de investigar esta questão serão conduzidos.

---

<sup>18</sup> BASILI et al., (1999) propõem a realização de famílias de experimentos para a geração de evidências em engenharia de software. A idéia de família de experiências é análoga, sendo que, ao invés de estudos experimentais, as organizações utilizariam pilotos e, ao invés de evidências em engenharia de software, as evidências geradas teriam significado restrito às organizações.

## REFERÊNCIAS BIBLIOGRÁFICAS

---

- AHMED-NACER, Mohamed, 2004, "Evolution of Software Processes and of their Models: A Multiple Strategy Approach", **Journal of Research and Practice in Information Technology**, Sydney, v. 36 (fev), pp. 9-22.
- AMBRIOLA, V.; CONRADI,R.; FUGGETTA, A., 1997, "Assessing process-centered software engineering environments", **ACM Transactions on Software Engineering and Methodology**, New York, v. 6 (jul.), pp. 283-328.
- ARDIMENTO, P. et al., 2003, "Empirical studies as a means for technology transfer. In: **Proceedings of the Workshop Series on Empirical Engineering**, 2., Roma, Italy,
- BALZER, R.; GRUHN, V., 2001, "Process-centered software engineering environments: academic and industrial perspectives". In: **Proceedings of the International Conference on Software Engineering**, 23., pp. 671 - 673, Toronto. Canadá.
- BASILI, V.R.; ROMBACH, H.D., 1987, "Tailoring the software process to project goals and environments." In: **Proceedings of the International Conference on Software Engineering**, 3., pp. 345 – 357.
- BASILI, V. R.; ROMBACH, H. D., 1988, "The TAME Project: Towards Improvement-Oriented Software Environments", **IEEE Transactions on Software Engineering**, v. 14 (jun.), pp. 758-773.
- BASILI, V. R., 1989, "Software development: a paradigm for the future". In: **Proceedings of the Annual International Computer Software and Applications Conference**, 13., pp. 471-485, Orlando, USA
- BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. "The Experience Factory", In: MARCINIAK, John J.. (Ed.). **Encyclopedia of Software Engineering**. New York: John Wiley & Sons, 1994. pp. 469-476.
- BASILI, V. R.; CALDIERA, G. "Improve Software Quality by Reusing Knowledge and Experience", **Sloan Management Review**, V. 37, N. 1, pp. 55-64, 1995.
- BASILI, V.; SHULL, F.; LANUBILE, F., 1999, "Building knowledge through families of experiments", **IEEE Transactions on Software Engineering**, V. 25, N. 4, pp. 456-473.

- BASILI, V. R. et al., 2002, "Lessons learned from 25 years of process improvement: The Rise and Fall of the NASA Software Engineering Laboratory". In: **Proceedings of the ICSE - International Conference on Software Engineering**, 24., pp. 69 – 79, Orlando. USA.
- BEITZ, A.; WIECZOREK, I., 2000, "Applying Benchmarking to Learn from Best Practices". In: **Proceedings of the PROFES - Product Focused Software Process Improvement**, 2, pp. 59 – 72, Oulu, Finland.
- BELL (Canadá). **The Trillium Model**. Copyright Bell Canada 1994. Disponível em: <<http://www.sqi.gu.edu.au/trillium/>>. Acesso em: 19 jan. 2005.
- BERGER, Patrícia Machado. **Instanciação de Processos de Software em Ambientes Configurados na Estação TABA**. Dissertação de Mestrado - Engenharia de Sistemas e Computação, PESC/COPPE/UFRJ. Rio de Janeiro, 2003.
- BIOLCHINI et al., 2005, **Systematic Review in Software Engineering**. Technical Report ES 679/05. Rio de Janeiro: PESC/COPPE/UFRJ.
- BRIAND, L.; EMAM, K. El; MELO, W. L., 1995, **AINSI An Inductive Method for Software Process Improvement: Concrete Steps and Guidelines**. Maryland, Computer Science Dep. University of Maryland.
- CAMPOS, F., 2005, **Melhoria de Processos e Evolução do Meta-ambiente da Estação TABA**. Exame de Qualificação (Doutorado) - Engenharia de Sistemas e Computação, Rio de Janeiro: PESC/COPPE/UFRJ.
- CAMPOS, F. et. al, 2005, **Estratégia em Níveis para a Avaliação e Melhoria de Processos de Software**. Relatório Técnico, Rio de Janeiro: PESC/COPPE/UFRJ.
- CONRADI, R.; FUGGETTA, A., 2002, "Improving Software Process Improvement", **IEEE Software**, v. 19, issue 4 (Jul./Aug.), pp. 92-99.
- CONRADI, R.; WANG, A. I., 2003, **Empirical Methods and Studies in Software Engineering - Experiences from ESERNET**. Berlin, Springer.
- CHRISISS, Mary Beth et al., 2003, **CMMI®: Guidelines for Process Integration and Product Improvement**. Boston, Addison Wesley.
- CONTE, T.; MENDES, E.; TRAVASSOS, G. H., 2005, "Processos de Desenvolvimento para Aplicações Web: Uma Revisão Sistemática". In: **Anais do Simpósio Brasileiro de Sistemas Multimídia e Web**, 11. Poços de Caldas, MG .

- CMU/SEI. *Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document*, CMU/SEI-2001-HB-001, Pittsburgh: Carnegie Mellon University/Software Engineering Institute, 2001. Disponível em: <<http://www.sei.cmu.edu/cmmi/appraisals/>>. Acesso em: 10 jan. 2006.
- CURTIS, B., 2000, "From MCC and CMM: technology transfers bright and dim". In: **Proceedings of the ICSE - International Conference on Software Engineering**, 22., pp. 521 – 530, Limerick, Ireland.
- DANTEKAR, A.; PERRY, D. E.; VOTTA, L. G., 1997, "Studies in Process Simplification", **Software Process: Improvement and Practice**, V. 3, N. 2 (Jun.), pp. 87-104.
- DEMING, W. E., 1982, **Out of the Crisis**. Study. MIT Center for Advanced Eng. Massachusetts, MIT Press.
- DUARTE, Elaine et al., 2005, "Uma Abordagem para Implantação de Processos de Software com ISO 9001 e CMMI" In: **Anais do SBQS - Simpósio Brasileiro de Qualidade de Software**, 4., pp. 403-410, Porto Alegre.
- DYBA, T.; KITCHENHAM, B.A.; JORGENSEN, M., 2005, "Evidence-based Software Engineering for Practitioners", **IEEE Software**, V. 22, N.1 (Jan./Feb.), pp. 58- 65.
- ELLMER, E., 1995, "Improving software processes". In: **Proceedings of the Conference on Software Engineering Environments**, 7., pp.74 – 83, Noordwijkerhout, Holanda.
- EIA. Electronic Industries Alliance Interim Standard (EIA/IS-731). 1998.
- FALBO, R., 1998, **Integração de Conhecimento em um Ambiente de Desenvolvimento de Software**, Tese (Doutorado) - Engenharia de Sistemas e Computação, PESC/COPPE/UFRJ, Rio de Janeiro.
- FARIAS, L. L., 2002, **Planejamento de Riscos em Ambientes de Desenvolvimento de Software Orientados à Organização**. Dissertação de Mestrado - Engenharia de Sistemas e Computação, PESC/COPPE/UFRJ. Rio de Janeiro.
- FOWLER, P.; RIFKIN, S., 1990, **Software Engineering Process Group Guide**. Pittsburgh, Software Engineering Institute.
- FENTON, N.; PFLEEGER, S.L.; GLASS, R.L., 1994, "Science and substance: a challenge to software engineers", **IEEE Software**, V. 11, N. 4 (Jul.), pp. 86-95.
- FOGLE, S.; LOULIS, C.; NEUENDORF, B., 2001, "The benchmarking process: one team's experience", **IEEE Software**, V.18, N. 5 (Sep-Oct), pp.40-47.

- FUGGETTA, A., 2000, "Software Process: A Roadmap." In: FINKELSTEIN, Anthony. (Ed.) **The Future of Software Engineering**, New York, ACM Press, p. 25-34.
- GARG, P. K.; JAZAYERI, M., 1996, "Process-centered software engineering environments: A grand tour", In: FUGGETTA, Alfonso; WOLF, Alexander. **Software Process**. New York, John Wiley & Sons, pp. 25-52.
- GREMBA, J.; MYERS, C. **The IDEAL Model: A Practical Guide for Improvement**. Disponível em: <<http://www.sei.cmu.edu/ideal/ideal.bridge.html>> Acesso em: 09 mai. 2006.
- HARRISON, W.; OSSHER, H.; TARR, P., 2000, "Software Engineering Tools and Environments: A Roadmap." In: **Proceedings of the International Conference on Software Engineering**, 22., pp. 261-277, Limerick, Ireland.
- HERBSLEB, J. D.; GOLDENSON, D. R., 1996, "A Systematic Survey of CMM Experience and Results". In: **Proceedings of the International Conference on Software Engineering**, 18., pp. 323 - 330, Berlin, Germany.
- HÖST, M. et al., 2001, "Exploring bottlenecks in market-driven requirements management processes with discrete event simulation", **Journal of Systems and Software**, V. 59, Issue 3 (dez.), pp. 323-332.
- IEEE, 1990, **IEEE STD 610.12-1990: Standard Glossary of Software Engineering Terminology**".
- ISO/IEC, 1995, **ISO/IEC 12207: Information Technology – Software Life-Cycle Processes**.
- ISO/IEC, 2003, **ISO/IEC 15504-1: Information Technology – Process Assessment. Part 1: Concepts and Vocabulary**.
- ISO/IEC, 2004, **ISO/IEC 15504-4: Information Technology – Process Assessment, Part 4: Guidance on use for Process Improvement and Process Capability Determination**.
- ISO/IEC, 2002, **ISO/IEC PDAM 12207: Information Technology – Amendment to ISO/IEC 12207**.
- ISO/IEC, 2004b, **ISO/IEC PDAM 12207: Information Technology – Amendment 2 to ISO/IEC 12207**.
- JEDLITSCHKA, A., 2005a, **Guidelines for Empirical Work in Software Engineering**. Technical Report, Kaiserslautern, Fraunhofer IESE.

- JEDLITSCHKA, A.; PFAHL, D., 2005, "Reporting Guidelines for Controlled Experiments". In: **Proceedings of the International Symposium on Empirical Software Engineering**, 4., pp. 92-101, Noosa Heads, Austrália.
- JEDLITSCHKA, A.; CIOLKOWSKI, M., 2004, "Towards evidence in software engineering". In: **Proceedings of the ISESE'04 - International Symposium on Empirical Software Engineering**, 3., pp. 261-270, Califórnia, USA.
- JEFFERY, R.; SCOTT, L., 2002, "Has Twenty-five Years of Empirical Software Engineering Made a Difference?" In: **Proceedings of the Asia-Pacific Software Engineering Conference**, 9., pp. 539-546, Gold Coast, Austrália.
- KARAHASANOVIC, A. et al., 2005, "Collecting Feedback During Software Engineering Experiment", **Empirical Software Engineering**, V. 10, Issue 2 (Apr.), pp. 113-147.
- KELLNER, M. I.; MADACHY, R. J; RAFFO, D. M., 1999, "Software process simulation modeling: Why? What? How?", **Journal of Systems and Software**, V. 46, Issues 2-3 (Apr.), p 91-105.
- KITCHENHAM, B.; PICKARD, L.; PFLEEGER, S.L., 1995, "Case studies for method and tool evaluation", **IEEE Software**, V. 12, Issue 4 (Jul.), pp. 52-62.
- KITCHENHAM, B.A.; PFLEEGER, S.L., 2002, "Principles of Survey Research Part 2: Designing a Survey", **Software Engineering Notes**, V. 27, N. 1 (Jan), pp. 18-20.
- KITCHENHAM, B.A. et al., 2002, "Preliminary Guidelines for Empirical Research in Software Engineering", **IEEE Transactions on Software Engineering**, V. 28, N. 8 (Aug.), pp. 721-734.
- KITCHENHAM, B., 2004, **Procedures for Performing Systematic Reviews**. Technical Report, Keele, Department of Computer Science Keele University.
- KULPA, M. K.; JOHNSON K. A., 2003, **Interpreting the CMMI: A Process Improvement Approach**. Florida, Auerbach Publications.
- MC FEELEY, B., 1996, **IDEAL<sup>SM</sup>: A User's Guide for Software Process Improvement**. Pittsburgh, Software Engineering Institute.
- MCGARRY, F. E.; THOMAS, M., 1994, "Top-Down vs. Bottom-Up Process Improvement", **IEEE Software**, V. 11, N. 7 (Jul.), pp. 12-13.
- MCGARRY, F. et al., 2002, "Attaining Level 5 in CMM process maturity". **IEEE Software**, V. 19, Issue 6 (Nov.-Dec.), pp. 87 – 96.

- MONTONI, M., et al., 2005, "Enterprise-Oriented Software Development Environments to Support Software Products and Processes Quality Improvement", **Lecture Notes of Computer Science (LNCS)**, pp. 370-384.
- MORISIO, M. , 1999, "Measurement processes are software, too", **Journal of Systems and Software**, V. 49, Issue 1 (Dec.), pp. 17-31.
- MÜNCH, J.; ARMBRUST, O., 2003, "Using Empirical Knowledge from Replicated Experiments for Software Process Simulation: A Practical Example." In: **Proceedings of the International Symposium on Empirical Software Engineering**, 2., pp.18 – 27, Roma, Italy.
- MÜNCH, J.; PFAHL, D.; RUS, I., 2005, "Virtual Software Engineering Laboratories in Support of Trade-off Analyses", **Software Quality Journal**, V.13, Issue 4 (Dec.), pp. 407-428.
- O'HARA, F., 2000, "European experiences with software process improvement". In: **Proceedings of the International Conference on Software Engineering**, 22., pp. 635-640, Limerick, Ireland.
- OLIVEIRA, K. M., 1999, **Modelo para Construção de Ambientes de Desenvolvimento de Software Orientados a Domínio**, Tese (Doutorado) - Engenharia de Sistemas e Computação, PESC/COPPE/UFRJ, Rio de Janeiro.
- OSTERWEIL, L., 1987, "Software Process Are Software Too." In: **Proceedings of the International Conference on Software Engineering**, 9., pp. 2-13. Monterey, USA.
- PAI, M. et al., 2004, "Systematic reviews and meta-analyses: An illustrated, step-by-step guide", **The National Medical Journal of India**, V. 17, Issue 2 (Mar-Apr), pp.86-95.
- PAULK, M. C. et al., 1993a. **Key practices of the capability maturity model, Version 1.1**. Pittsburgh, Software Engineering Institute.
- PAULK, M. C. et al., 1993b, "Capability maturity model, version 1.1", **IEEE Software**, V.10, N. 4 (Jul.), pp.18-27.
- PFAHL, D.; BIRK, A., 2000, "Using Simulation to Visualise and Analyse Product-Process Dependencies in Software Development Projects", **Lecture Notes in Computer Science**, V. 1840 (Jan.), p. 88.

- PFAHL, D.; LEBSANFT, K., 2000, "Using simulation to analyse the impact of software requirement volatility on project performance", **Information and Software Technology**, V. 42, Issue 14 (Nov), pp. 1001-1008.
- PFAHL, D.; RUHE, G., 2002, "IMMoS - A methodology for integrated measurement, modelling, and simulation", **Software Process Improvement and Practice**, V. 7 (Dez.), pp. 189–210.
- PFLEEGER, S. L., 1999, "Albert Einstein and Empirical Software Engineering", **IEEE Computer**, V. 32, Issue 10 (Oct.), pp. 32-37.
- PFLEEGER, S. L., 2004, **Engenharia de Software – Teoria e Prática**, 2ª São Paulo, Prentice Hall.
- PFLEEGER, S.L.; KITCHENHAM, B.A., 2001, "Principles of Survey Research - Part 1: Turning Lemons into Lemonade", **Software Engineering Notes**, V. 26, N. 6 (Nov.), pp. 16-18.
- RAFFO, D. , 1993, "Evaluating the impact of process improvements quantitatively using process modeling". In: **Proceedings of the Conference of the Centre for Advanced Studies on Collaborative Research**, 3., pp. 290-313. Toronto, Canadá.
- RIFKIN, S., 2001, "Why Software Process Innovations Are Not Adopted", **IEEE Software**, V. 18, Issue 4 (Jul.-Aug.), pp. 110-112.
- ROCHA, A. R. C.; AGUIAR, T. C.; SOUZA, J. M., 1990, "Taba: A Heuristic Workstation for Software Development". In: **Proceedings of the COMPEURO '90, IEEE INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING**, pp. 126-129, Tel Aviv, Israel.
- ROCHA, A. R.; MALDONADO, J. C.; WEBER, K. C., 2001, **Qualidade de Software – Teoria e Prática**, 1ª ed., São Paulo, Prentice Hall.
- SANTOS, G., 2005, "Using a Software Development Environment with Knowledge Management to Support Deploying Software Processes in Small and Medium Size Companies". **Workshop Learning Software Organization**, pp. 72-76, Apr.
- SILVA FILHO, R. C.; ROCHA, A. R.; TRAVASSOS, G. H., 2005, "Uma Abordagem Experimental para Avaliação da Melhoria de Processos". In: **Anais do Simpósio Brasileiro de Qualidade de Software**, 4., pp. 395 – 402, Porto Alegre.
- SOFTEX, 2005, **MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral Versão 1.0**, Campinas, SP, 2005. 61 p.

- SHULL, F.; CARVER, J.; TRAVASSOS, G. H., 2001, "An Empirical Methodology for Introducing Software Processes", **ACM SIGSOFT Software Engineering Notes**, V. 26, Issue 5 (Sep.), pp. 288 – 296.
- SHULL, F. et al., 2004, "Knowledge-Sharing Issues in Experimental Software Engineering", **Empirical Software Engineering: An International Journal**, V. 9, Issue. 1 (Mar.), pp.111-137.
- SJØBERG, D. I. K. et al., 2002, "Conducting realistic experiments in software engineering". In: **Proceedings of the International Symposium on Empirical Software Engineering**, 1., pp. 17- 26. Nara, Japan, Oct.
- SJØBERG, D. I. K. et al., 2005, "A Survey of Controlled Experiments in Software Engineering", **IEEE Transactions on Software Engineering**, V.31, N. 9 (Sep.), pp. 733-753.
- SOLINGEN, R. VAN; BERGHOUT, E., 1999, **The Goal/Question/Metric Method: a practical guide for quality improvement of software development**. London, McGraw-Hill.
- TRAVASSOS, G. H., 1994, **O Modelo de Integração de Ferramentas da Estação TABA**, Tese (Doutorado) - Engenharia de Sistemas e Computação, PESC/COPPE/UFRJ, Rio de Janeiro.
- TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. A., 2002, **Introdução à Engenharia de Software Experimental**. Relatório Técnico ES 590/02. Rio de Janeiro, PESC/COPPE/UFRJ.
- TVEDT, J. D.; COLLOFELLO, J. S., 1995, "Evaluating the effectiveness of process improvements on software development cycle time via system dynamics modeling". In: **Proceedings of the Annual International Computer Software and Application Conference**, pp. 318-325, Dallas, USA.
- VILLELA, K., 2004, **Definição e Construção de Ambientes de Desenvolvimento de Software Orientados à Organização**. 2004. Tese (Doutorado) - Engenharia de Sistemas e Computação, PESC, COPPE/UFRJ, Rio de Janeiro.
- VILLELA, K. et al., 2004, "Definição de Processos em Ambientes de Desenvolvimento de Software Orientados a Organização". In: **Anais do Simpósio Brasileiro de Qualidade de Software**, pp. 4-17, Brasília.
- ZELKOWITZ, M.V., 1996, "Software engineering technology infusion within NASA", **IEEE Transactions on Engineering Management**, V. 43, N.3 (Aug.), pp.250-261.

- ZELKOWITZ, M. V.; WALLACE, D. R., 1998, "Experimental Models for Validating Technology", **IEEE Computer**, V. 31, Issue 5 (May), pp.23 – 31.
- ZELKOWITZ, M. V.; WALLACE, D. R.; BINKLEY, D. W., 2003, "Experimental Validation of New Software Technology", **Lecture Notes on Empirical Software Engineering**, **World Scientific Publishing**, pp. 229-263.
- WALRAD, C.; MOSS, E., 1993, "Measurement: the key to application development quality", **IBM Systems Journal**, V. 32, Issue. 3 (Sep.), pp. 445-460.
- WEBER, K. C. et al., 2004, "Modelo de Referência para Melhoria de Processo de Software: uma abordagem brasileira". In: **Anais da Conferência Latino-Americana de Informática**, 30., pp. 461-476, Arequipa, Peru.
- WEBER, K. C. et al., 2005, "Brazilian Software Process Reference Model and Assessment Method", P. Yolum et al. (Eds.). **Lecture Notes in Computer Science**, V. 3733, pp. 402-411.
- WERNECK, V. M., 1995, **Ambiente para Desenvolvimento de Software Baseado em Conhecimento**, Tese (Doutorado) - Engenharia de Sistemas e Computação, PESC/COPPE/UFRJ, Rio de Janeiro.
- WOHLIN, C. et. al., 2000, **Experimentation in Software Engineering – An Introduction**. Massachusetts, Kluwer Academic Publishers.

# APÊNDICE A - Protocolo do estudo baseado em revisão sistemática

---

## 1. Contexto

Modelos e normas em prol da qualidade sugerem o uso de pilotos para testar a efetividade das propostas de melhoria de processos de software que apresentam algum risco e/ou incertezas associadas à sua institucionalização. A partir da norma ISO/IEC 15504 e dos modelos CMMI, IDEAL e QIP, foram extraídas práticas que visam potencializar os resultados obtidos na realização de pilotos:

- a) Uso de indicadores: Os indicadores podem ser quantitativos ou qualitativos. Quantitativos são obtidos a partir de métricas, que são selecionadas previamente para prover uma descrição quantitativa dos efeitos observados. Os indicadores qualitativos são oriundos de avaliações de especialistas, post-mortem, dentre outras;
- b) Planejamento: O piloto tem um propósito específico e, para atingir seus objetivos, é necessário alocar recursos e tomar todas as providências necessárias para que o piloto seja executado de forma permitir a observação dos efeitos das mudanças no processo.
- c) Seleção da Equipe: Os participantes devem ser pessoas dispostas e capazes de aprender e executar as ações de melhoria propostas.
- d) Treinamento da Equipe: A equipe deve ser treinada para que as ações propostas sejam executadas conforme descritas.
- e) Uso de mais de um projeto: O número de projetos pode aumentar a visibilidade dos efeitos e dos fatores que exerceram influência nos resultados, contribuindo com o refinamento da proposta de melhoria.
- f) Registro de Comprometimento: Obter o comprometimento das pessoas é um dos fatores de sucesso em qualquer projeto. No caso dos pilotos, requer uma atenção especial, pois em geral exige mudanças no desempenho das atividades.
- g) Adequação do Ambiente de Operação: Um piloto executado em um ambiente diferente daquele que os projetos são realizados pode não estar sujeito a todas as intempéries do cotidiano da organização e, conseqüentemente, a avaliação poderá retratar um resultado distorcido.
- h) Execução monitorada: É preciso verificar se durante a operação o ambiente está conforme o planejado e se as medidas estão sendo coletadas adequadamente.
- i) Avaliação dos resultados: Análise quantitativa e/ou qualitativa dos resultados obtidos com o piloto.
- j) Documentação dos Resultados: Ao final, todo conhecimento explicitado (problemas encontrados e lições aprendidas) pelo piloto deve ser devidamente registrado e disponibilizado para a organização.

Esses modelos em prol da qualidade estão restritos a sugestão de práticas, ou seja, eles não fornecem instrumentos (procedimentos, processos, abordagens, técnicas, métodos,

ferramentas ou qualquer outro mecanismo) que apóiem a condução de projetos-piloto de forma acurada.

Portanto, com o objetivo de identificar instrumentos que facilitem a avaliação das propostas de melhoria de processos de software no ambiente industrial através da realização de projetos-piloto, propõe-se a execução deste estudo, baseado em revisão sistemática, com o intuito de identificar e caracterizar tais instrumentos.

## **2. Questão de pesquisa**

### **2.1. Questão principal**

Quais instrumentos têm sido propostos para facilitar a realização de pilotos no âmbito industrial com o objetivo de avaliar a efetividade das mudanças dos processos de software em prol da melhoria?

### **2.2. Questões secundárias**

Uma vez que os instrumentos sejam identificados, propõe-se uma caracterização baseada nas práticas sugeridas pelos modelos e normas em prol da qualidade que sugerem a adoção de pilotos para avaliar a efetividade das propostas de melhoria.

Para caracterizar esses instrumentos propõem-se as seguintes questões:

- Quais tipos de instrumentos<sup>19</sup> estão sendo propostos?
- Quais práticas, propostas pelos modelos e normas em prol da qualidade, são atendidas pelos instrumentos?
- Quais instrumentos tiveram seu uso avaliado na indústria?
- Quais instrumentos fornecem ou utilizam ferramentas de apoio?
- Qual o modelo ou norma em prol da qualidade mais referenciada por estes instrumentos?

## **3. População**

Trabalhos publicados em conferências e periódicos propondo ou relatando o uso de pilotos para melhoria de processos de software.

## **4. Intervenção**

Realização de pilotos para avaliação da efetividade das propostas de melhoria de processos de software no âmbito industrial.

## **5. Resultados**

Espera-se identificar um conjunto de instrumentos que apóiam a avaliação da efetividade das propostas de melhorias através da realização de projetos-pilotos na indústria de software.

---

<sup>19</sup> Neste caso, instrumentos correspondem a procedimentos, processos, abordagens, técnicas, métodos, ferramentas ou quaisquer mecanismos que facilitem a realização de pilotos.

## **6. Abordagem de Avaliação e Experimentação**

Não será feita qualquer avaliação no que diz respeito à validade das propostas ou experiências relatadas. Ou seja, o interesse é restrito à identificação e caracterização dos instrumentos com relação aos modelos e normas em prol da qualidade.

A premissa é que a realização de pilotos para avaliar a efetividade das propostas de melhoria de processo de software, dotada por práticas sugeridas pelos modelos e normas em prol da qualidade, possui grandes chances de prover resultados com algum significado.

## **7. Escopo da pesquisa**

Para delinear o escopo da pesquisa foram estabelecidos critérios para garantir, de forma equilibrada, a viabilidade da execução (custo, esforço e tempo), acessibilidade aos dados (disponibilidade das publicações) e abrangência do estudo.

A pesquisa dar-se-á a partir de buscas em bibliotecas digitais, através dos seus respectivos engenhos de busca.

Critérios adotados para seleção das bibliotecas:

- Possuir engenho de busca que permita o uso de expressões lógicas ou mecanismo equivalente;
- Pertencer a uma das editoras listadas no Portal de Periódicos da CAPES;
- Incluir em sua base publicações da área de exatas ou correlatas que possuam relação direta com o tema a ser pesquisado;
- Os engenhos de busca deverão permitir a busca no texto completo das publicações.

A pesquisa está restrita a análise de publicações obtidas, exclusivamente, a partir das bibliotecas digitais. O acesso a estas publicações não deve incorrer em ônus para a pesquisa.

## **8. Fontes**

O estudo englobará todas as publicações das editoras selecionadas conforme os critérios definidos no escopo da pesquisa e publicadas até 31 de dezembro de 2005. Não há restrições referentes à data inicial de publicação nas fontes de informação.

## **9. Idioma**

Para a realização desta pesquisa o idioma selecionado foi o inglês. A escolha do idioma deve-se à adoção do idioma pela grande maioria das conferências e periódicos relacionados como tema de pesquisa e por ser o idioma utilizado pela maioria das editoras relacionadas com o tema listadas no Portal de Periódicos da CAPES.

## **10. Expressão de busca**

“software process improvement” AND pilot

## 11. Procedimentos de seleção e critérios

### 11.1. Procedimentos de seleção

A seleção dos estudos dar-se-á em três etapas:

i) **Seleção e catalogação preliminar das publicações.** A seleção preliminar das publicações será feita a partir da aplicação da expressão de busca nos engenhos providos pelas fontes selecionadas. Cada publicação será catalogada em um banco de dados criado especificamente para este fim e armazenada em um repositório para análise posterior;

ii) **Seleção das publicações relevantes - 1º filtro.** A seleção preliminar com o uso da expressão de busca não garante que todo o material coletado seja útil no contexto da pesquisa, pois a aplicação de expressões de busca em engenhos é restrita ao aspecto sintático. Para excluir os documentos irrelevantes contidos no conjunto preliminar, devem-se aplicar os seguintes critérios:

- **Menção à alteração de processos de software.** A publicação retrata alguma alteração realizada em processos de software em prol da melhoria?
- **Uso de pilotos para avaliar a efetividade das alterações em processos de software.** A publicação ilustra ou recomenda o uso de pilotos para testar a efetividade das alterações realizada nos processos de software?

As publicações que atendem a esses critérios serão movidas para um segundo repositório que será utilizado na terceira etapa da seleção.

iii) **Seleção das publicações relevantes - 2º filtro.** Boa parte dos documentos que compõem o repositório obtido a partir da aplicação do primeiro filtro podem não conter dados a serem coletados. Para minimizar o esforço de coleta, um segundo filtro deve ser aplicado para originar um repositório contendo apenas documentos que possuem dados a serem coletados. Critério a ser aplicado:

- **Instrumentos para apoiar a realização de pilotos no ambiente industrial.** A publicação deve apresentar algum instrumento<sup>20</sup> que apóia a realização de pilotos na avaliação da efetividade das propostas de melhoria em um ambiente industrial. Os instrumentos identificados deverão ser cadastrados no banco de dados relacionando-o com a publicação, tipo de instrumento, o contexto do trabalho e o modelo de referência utilizado.

Para diminuir o risco que uma publicação seja excluída prematuramente em uma das etapas do estudo, sempre que existir dúvida, a publicação não deverá ser excluída.

---

<sup>20</sup> Processo, abordagem, procedimentos, ferramentas, técnicas, métodos ou qualquer mecanismo que auxilie o planejamento, execução, monitoramento ou análise do piloto.

## 11.2. Critérios de exclusão

Não serão consideradas as publicações que:

- O uso do conceito de pilotos não esteja relacionado à avaliação da efetividade das propostas de melhoria de processos de software;
- Apresentem instrumentos para a avaliação da efetividade das propostas de melhoria de processos de software através que não apóiam o uso de pilotos;

## 12. Procedimentos para Extração dos Dados

### 12.1. Catalogação Preliminar

Armazenamento da referência completa fornecida pelo engenho de busca com indicação de fonte e biblioteca ao qual foi coletada.

### 12.2. 1º Filtro – Seleção das publicações relevantes

Cada publicação contida na catalogação preliminar deve ser examinada com o objetivo de responder as questões de seleção. Cada publicação que contiver respostas positivas para os critérios de seleção deverá ter seu registro marcado como “verificada na primeira seleção, passou”, do contrário, o registro deverá ser marcado como “verificada na primeira seleção, não passou”.

### 12.3. 2º Filtro – Seleção das publicações relevantes

Cada publicação marcada como “verificada na primeira seleção, passou” deverá ser analisada com relação ao critério do segundo filtro. Todas aquelas que atenderem aos critérios do segundo filtro, deverão ser marcadas como “verificada na segunda seleção, passou”, do contrário deverão ser marcadas como “verificada na segunda seleção, não passou”.

Cada publicação selecionada no segundo filtro será submetida a uma extração preliminar dos dados. Estes dados deverão ser armazenados em um banco de dados e devem conter: a classificação da publicação (relato de experiência, proposta, avaliação experimental); tipo de instrumento (conjunto de procedimentos, processo, abordagem, método, técnica ou outros); o modelo ou norma que serviu de referência para construção do instrumento; e um breve resumo.

### 12.4. Extração de Dados

Todas as publicações selecionadas no segundo filtro e devidamente catalogadas serão submetidas ao procedimento de extração de dados com relação às questões secundárias do estudo, que consiste em: verificar quais práticas sugeridas por modelos e normas em prol da qualidade que o instrumento atende, se o instrumento teve seu uso avaliado na indústria e se o instrumento fornece ou utiliza ferramentas de apoio.

## **13. Procedimentos para Análise**

### **13.1. Análise Quantitativa**

A análise quantitativa dar-se-á pela extração direta dos dados a partir do banco de dados com os registros dos achados.

A análise quantitativa consiste em fornecer:

- O número de instrumentos identificados por tipo.
- Total de instrumentos que atendem cada prática atendida
- Instrumentos que utilizam apoio ferramental para apoiar a execução das boas práticas
- Total de instrumentos por classificação
- Total de instrumentos por modelo de referência

### **13.1. Análise Qualitativa**

A análise qualitativa deve utilizar como base os dados quantitativos e realizar considerações com o intuito de construir uma visão ampla com relação a diversidade de instrumentos, a aplicação destes instrumentos em um ambiente industrial e a relação dos instrumentos com as práticas sugeridas pelos modelos e normas em prol da qualidade.

## APÊNDICE B - Fontes das publicações identificadas na seleção preliminar

Fontes	Total de Publicações
ACM Conference on Hypertext and Hypermedia	1
ACM SIGAda Ada Letters	3
ACM SIGCPR Conference on Computer Personnel Research	1
ACM SIGICE Bulletin	1
ACM SIGMIS Database	1
ACM SIGSOFT Software Engineering Notes	5
ACM Symposium on Applied Computing	1
Aerospace	1
Agile Development Conference	1
Annals of Software Engineering	1
Annual Frontiers in Education	1
Annual Hawaii International Conference on System Sciences	24
Annual International Computer Software and Applications Conference	4
Annual International Conference on Computer documentation	1
Annual International Workshop on Software Technology and Engineering Practice	1
Annual Quality Congress	1
ASLIB	2
Automated Software Engineering	1
Business Process Management Journal	2
Centre for Advanced Studies on Collaborative research: software engineering	5
CHI '02 extended abstracts on Human factors in computing systems	1
Communications of the ACM	8
Computer	10
Computerworld	1
Conference on Australian Computing Education	1
Conference on Software Engineering Education and Training	4
Database for Advances in Information Systems	1
Decision Support Systems	1
Design Management Journal	1
Empirical Software Engineering	12
Euromicro Conference	3
European Conference on Software Maintenance and Reengineering	1
European Software Engineering Conference	2
Expert Systems with Applications	1
Human Systems Management	1
IBM Systems Journal	2
IEE Colloquium on Marketing Software Engineering Marketing Software Engineering	1
IEE Colloquium on Project Management for Software Engineers	1
IEE Proceedings Software	1
IEEE International Conference on Information Reuse and Integration	1
IEEE International Conference on Systems	3
IEEE Joint International Conference on Requirements Engineering	1
IEEE Journal on Selected Areas in Communications	1
IEEE Nuclear Science Symposium Conference Record	1
IEEE Security & Privacy Magazine	1
IEEE Software	34
IEEE Std 1348-1995	1
IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part A	1
IEEE Transactions on Education	1
IEEE Transactions on Engineering Management	4
IEEE Transactions on Software Engineering	12
IEEE Transactions on Systems, Man and Cybernetics	1
IEEE Workshop on Future Trends of Distributed Computing Systems	1
Industrial Management & Data Systems	2
Information & Management	1
Information and Software Technology	5
Information Management & Computer Security	1
Information Resources Management Journal	1
Information Systems Frontiers	2
Information Systems Journal	1
Information Technology & People	2
InfoWorld	2
International Conference on Artificial intelligence and Law	1
International Conference on Machine Learning and Cybernetics, 2003	1
International Conference on Quality Software	4
International Conference on Software Engineering	31
International Conference on Software Maintenance	6
International Conference Software Engineering: Education and Practice	1
International Requirements Engineering Conference	1
International Symposium and Forum on Software Engineering Standards	2
International Symposium Empirical Software Engineering	14

International Symposium on Software Metrics	8
International Symposium on Software Reliability Engineering	1
IT Professional	2
Journal of Database Management	2
Journal of Information Technology Case and Application Research	2
Journal of Management Information Systems	2
Journal of Systems and Software	11
Lecture Notes in Computer Science	20
Management Science	1
Managerial Auditing Journal	2
Mexican International Conference on Computer Science	1
MIS Quarterly	5
NASA Goddard Software Engineering Workshop	1
Organization	1
Portland International Conference on Management and Technology Innovation in Technology Management (PICMET)	2
Quality Progress	1
Requirements Engineering	1
Software Engineering Conference	4
Software Quality Journal	15
The AIAA/IEEE/SAE Digital Avionics Systems Conference	3
The Journal of Business Strategy	1
Training for Quality	2
World Class Design to Manufacture	1

## APÊNDICE C - Publicações obtidas a partir de mais de uma biblioteca digital

Publicações duplicadas na seleção preliminar	Bibliotecas
Achieving industrial relevance with academic excellence: lessons from the Oregon master of software engineering	IEEE Xplore ACM Digital Library
An evaluation of TQM and the techniques for successful implementation	ProQuest Emerald Insight
Applying and adjusting a software process improvement model in practice: the use of the IDEAL model in a small software enterprise	ACM Digital Library IEEE Xplore
Combining quality and software improvement	ProQuest ACM Digital Library
Elements of a Realistic CASE Tool Adoption Budget	ACM Digital Library ProQuest
European experiences with software process improvement	IEEE Xplore ACM Digital Library
Experiences in the Application of Software Process Improvement in SMES	ProQuest SpringerLink
From MCC to CMM: technology transfers bright and dim	IEEE Xplore ACM Digital Library
Introduction of modern software technology. A case from Swedish industry	ProQuest SpringerLink
On the limitations of software process assessment and the recognition of a required re-orientation for global process improvement	SpringerLink ProQuest
Perceptions of software quality: a pilot study	ProQuest SpringerLink
Process improvement in facilities management: the SPICE approach	Emerald Insight ProQuest
Product Focused Software Process Improvement: Concepts and Experiences from Industry	ProQuest SpringerLink
Quality through Managed Improvement and Measurement (QMIM): Towards a Phased Development and Implementation of a Quality Management System for a Software Company	ProQuest SpringerLink
Reinventing methodology: who reads it and why?	ProQuest ACM Digital Library
Second Workshop on Software Quality	ACM Digital Library IEEE Xplore
Situated assessment of problems in software development	ACM Digital Library ProQuest
Six Sigma in the software industry: results from a pilot study	Emerald Insight ProQuest
Software process improvement: making it happen in practice	ProQuest SpringerLink
Software Process Models and Project Performance	SpringerLink ProQuest
Software quality management and ISO 9000 implementation	ProQuest Emerald Insight
Teams in packaged software development: The Software Corp. experience	Emerald Insight ProQuest
Technology transfer macro-process: a practical guide for the effective introduction of technology	IEEE Xplore ACM Digital Library
The evolution of goal-based information modelling: literature review	ProQuest Emerald Insight
Virtual Software Engineering Laboratories in Support of Trade-off Analyses	ProQuest SpringerLink

## APÊNDICE D - Publicações selecionadas para filtragem

Id.	Título	Autoria	DadosPublicacao
1	"You can't measure us, we do client/server." Dispelling developers' myths about software measurement"	Carol A Dekkers	Quality Congress. ASQ's ... Annual Quality Congress Proceedings. 1998. pág. 307 (9 páginas)
2	1998 IEEE International Conference on Systems, Man, and Cybernetics	IEEE	Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on , vol.5, no.pp.0_2-0_56, 11-14 Oct 1998
3	1998 IEEE International Conference on Systems, Man, and Cybernetics - vol. 3 of 5	IEEE	Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on , vol.3, no.pp.0_1-1_31, 11-14 Oct 1998
4	A case study of software process improvement during development	Bhandari, I.; Halliday, M.; Tarver, E.; Brown, D.; Chaar, J.; Chillarege, R.	Software Engineering, IEEE Transactions on , vol.19, no.12pp.1157-1170, Dec 1993
5	A family of experiments to validate metrics for software process models	G. Canfora, F. Garcia, M. Piattini, F. Ruiz and C.A. Visaggio	Journal of Systems and Software, Volume 77, Issue 2, August 2005, Pages 113-129
6	A field study of the adoption of software process innovations by information systems professionals	Agarwal, R.; Prasad, J.	IEEE Transactions on Engineering Management, vol.47, no.3pp.295-308, Aug 2000
7	A framework for assessing the use of third-party software quality assurance standards to meet FDA medical device software process control guideline's	Bovee, M.W.; Paul, D.L.; Nelson, K.M.	Engineering Management, IEEE Transactions on , vol.48, no.4pp.465-478, Nov 2001
8	A framework for assisting the design of effective software process improvement implementation strategies	Mahmood Niazi, David Wilson and Didar Zowghi	Journal of Systems and Software, Volume 78, Issue 2, November 2005, Pages 204-222
9	A Function Point-Like Measure for Object-Oriented Software	Giuliano Antoniol, Chris Lokan, Gianluigi Caldiera, Roberto Fiutem	Empirical Software Engineering, Volume 4, Issue 3, Sep 1999, Pages 263 - 287
10	A Meta-model Framework for Software Process Modeling	Marcello Visconti, Curtis R. Cook	Lecture Notes in Computer Science, Volume 2559, Jan 2002, Pages 532 - 545
11	A method for software quality planning, control, and evaluation	Boegh, J.; Depanfilis, S.; Kitchenham, B.; Pasquini, A.	Software, IEEE , vol.16, no.2pp.69-77, Mar/Apr 1999
12	A model for the implementation of software process improvement: a pilot study	Niazi, M.; Wilson, D.; Zowghi, D.	Quality Software, 2003. Proceedings. Third International Conference on , vol., no.pp. 196- 203, 6-7 Nov. 2003
13	A Multi-variant Approach to Software Process Modelling	Wolfgang Hesse, Jörg Noack	Lecture Notes in Computer Science, Volume 1626, Jan 1999, Page 210
14	A procedure for customizing a software process	Ibarguengoitia, G.; Salazar, J.A.; Sanchez, M.G.; Ramirez, A.Y.	Computer Science, 2003. ENC 2003. Proceedings of the Fourth Mexican International Conference on , vol., no.pp. 68- 72, 8-12 Sept. 2003
15	A process for evaluating legal knowledge-based systems based upon the context criteria contingency-guidelines framework	María Jean J. Hall, Richard Hall, John Zeleznikow	Proceedings of the 9th international conference on Artificial intelligence and law, Scotland, United Kingdom, Pages: 274 - 283, 2003
16	A quality doctrine for software: do it right the first time	Cross, S.E.	Software Engineering Conference, 2002. Ninth Asia-Pacific , vol., no.pp. 187- 194, 2002
17	A single model for process improvement	Ibrahim, L.; Pyster, A.	IT Professional , vol.6, no.3pp. 43- 49, May-June 2004
18	A software defect report and tracking system in an intranet	Monteiro, A.; Almeida, A.B.; Goulao, M.; Abreu, F.B.; Sousa, P.	Software Maintenance and Reengineering, 1999. Proceedings of the Third European Conference on , vol., no.pp.198-201, 1999
19	A software development process for small projects	Russ, M.L.; McGregor, J.D.	Software, IEEE , vol.17, no.5pp.96-101, Sep/Oct 2000
20	A survey of controlled experiments in software engineering	Sjoeborg, D.I.K.; Hannay, J.E.; Hansen, O.; Kampenes, V.B.; Karahasanovic, A.; Liborg, N.-K.; Rekdal, A.C.	Software Engineering, IEEE Transactions on , vol.31, no.9pp. 733- 753, Sept. 2005
21	A survey on software estimation in the Norwegian industry	Moloecken-Oestvold, K.; Joergensen, M.; Tanilkan, S.S.; Gallis, H.; Lien, A.C.; Hove, S.W.	Software Metrics, 2004. Proceedings. 10th International Symposium on , vol., no.pp. 208- 219, 14-16 Sept. 2004
22	A survey on software reuse processes	de Almeida, E.S.; Alvaro, A.; Lucredio, D.; Garcia, V.C.; de Lemos Meira	Information Reuse and Integration, Conf, 2005. IRI - 2005 IEEE International Conference on , vol., no.pp. 66- 71, 15-17 Aug. 2005
23	A survey on the Software Maintenance Process	Sousa, M.J.C.; Moreira, H.M.	Software Maintenance, 1998. Proceedings. International Conference on , vol., no.pp.265-274, 16-20 Nov 1998
24	A systematic survey of CMM experience and results	James D. Herbsleb, Dennis R. Goldenson	Proceedings of the 18th international conference on Software engineering, Berlin, Germany, Pages: 323 - 330, 1996
25	A Systems Perspective on Software Process Improvement	Andreas Birk, Dietmar Pfahl	Lecture Notes in Computer Science, Volume 2559, Jan 2002, Page 4

26	A variability management process for software product lines	Edson Alves de Oliveira, Itana M. S. Gimenes, Elisa Hatsue Moriya Huzita, José Carlos Maldonado	Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada, Pages: 225 - 241, 2005
27	Acceptance Issues in Metrics Program Implementation	Umarji, M.; Emurian, H.	Software Metrics, 2005. 11th IEEE International Symposium , vol., no.pp. 20- 20, 19-22 Sept. 2005
28	Achieving industrial relevance with academic excellence: lessons from the Oregon master of software engineering	Faulk, S.R.	Software Engineering, 2000. Proceedings of the 2000 International Conference on , vol., no.pp.293-302, 2000
29	Activity-based OO business modeling and control	Succi, G.; Benedicenti, L.; De Panfilis, S.; Vernazza, T.	IT Professional , vol.2, no.3pp.45-50, May/June 2000
30	Adopting a software security improvement program	Taylor, D.; McGraw, G.	Security & Privacy Magazine, IEEE , vol.3, no.3pp. 88- 91, May-June 2005
31	Adopting the SW-CMM in a small IT organization	Guerrero, F.; Eterovic, Y.	Software, IEEE , vol.21, no.4pp. 29- 35, July-Aug. 2004
32	Aggregating viewpoints for strategic software process improvement-a method and a case study	Karlstrom, D.; Runeson, P.; Wohlin, C.	Software, IEE Proceedings- [see also Software Engineering, IEE Proceedings] , vol.149, no.5pp. 143- 152, Oct 2002
33	Agile software development in large organizations	Lindvall, M.; Muthig, D.; Dagnino, A.; Wallin, C.; Stupperich, M.; Kiefer, D.; May, J.; Kahkonen, T.	Computer , vol.37, no.12pp. 26- 34, Dec. 2004
34	Aligning strategies: organizational, project, individual [IT governance]	Hefner, R.	System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on , vol., no.pp. 9 pp.-, 6-9 Jan. 2003
35	ami: a case study in promoting successful technology transfer	Rowe, A.	Marketing Software Engineering, IEE Colloquium on , vol., no.pp.3/1-3/6, 19 Jan 1996
36	An architecture for integrating concurrency control into environment frameworks	George T. Heineman, Gail E. Kaiser	Proceedings of the 17th international conference on Software engineering, Seattle, Washington, United States, Pages: 305 - 313, 1995
37	An early SPICE experience	Han van Loon	Quality Progress. Milwaukee: Feb 2000. Vol. 33, Num. 2; pág. 99 (6 páginas)
38	An empirical analysis of customer satisfaction for Intranet marketing systems	Mayuram S. Krishnan and Venkatram Ramaswamy	Decision Support Systems, Volume 24, Issue 1, November 1998, Pages 45-54
39	An empirical analysis of productivity and quality in software products	M S Krishnan, C H Kriebel, Sunder Kekre, Tridas Mukhopadhyay	Management Science. Linthicum: Jun 2000. Vol. 46, Num. 6; pág. 745
40	An empirical evaluation of the G/Q/M method	Khaled El Emam, Nadir Moukheiber, Nazim H. Madhavji	Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: software engineering - Volume 1, Toronto, Ontario, Canada, Pages: 265 - 289, 1993
41	An empirical investigation of the key factors for success in software process improvement	Dyba, T.	Software Engineering, IEEE Transactions on , vol.31, no.5pp. 410- 424, May 2005
42	An empirical study of groupware support for distributed software architecture evaluation process	Muhammad Ali Babar, Barbara Kitchenham, Liming Zhu, Ian Gorton and Ross Jeffery	Journal of Systems and Software, In Press, Corrected Proof, Available online 2 February 2006
43	An Empirical Study of Software Process in Practice	Coleman, G.	System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on , vol., no.pp. 315c- 315c, 03-06 Jan. 2005
44	An empirical study on the utility of formal routines to transfer knowledge and experience	Reidar Conradi, Tore Dybå	Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering, Vienna, Austria, Pages: 268 - 276, 200
45	An evaluation of applying use cases to construct design versus validate design	Syversen, E.; Anda, B.; Sjeberg, D.I.K.	System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on , vol., no.pp. 10 pp.-, 6-9 Jan. 2003
46	An evaluation of TQM and the techniques for successful implementation	Peter Kia Liang Tan	Training for Quality. Bradford: 1997. Vol. 5, Num. 4; pág. 150
47	An examination of the current state of IPSE technology	Alan W. Brown	Proceedings of the 15th international conference on Software Engineering, Baltimore, Maryland, United States, Pages: 338 - 347, 1993
48	An experience on using the team software process for implementing the Capability Maturity Model for software in a small organization	Serrano, M.A.; de Oca, C.M.; Cedillo, K.	Quality Software, 2003. Proceedings. Third International Conference on , vol., no.pp. 327- 334, 6-7 Nov. 2003
49	An experiment on software project size and effort estimation	Passing, U.; Shepperd, M.	Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on , vol., no.pp. 120- 129, 30 Sept.-1 Oct. 2003
50	An experimental evaluation of inspection and testing for detection of design faults	Andersson, C.; Thelin, T.; Runeson, P.; Dzamashvili, N.	Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on , vol., no.pp. 174- 184, 30 Sept.-1 Oct. 2003
51	An expert system in the domain of software technology transfer	P. FowlerI. García MartínN. JuristoL. LevineJ. L. Morant	Expert Systems with Applications, Volume 12, Issue 3, April 1997, Pages 275-300
52	An Industrial Case Study of Implementing and Validating Defect Classification for Process Improvement and Quality Management	Freimut, B.; Denger, C.; Ketterer, M.	Software Metrics, 2005. 11th IEEE International Symposium , vol., no.pp. 19- 19, 19-22 Sept. 2005

53	An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software	Björn Regnell, Martin Höst, Johan Natt och Dag, Per Beremark, Thomas Hjelm	Requirements Engineering, Volume 6, Issue 1, Feb 2001, Pages 51 - 62
54	An Instrument for Assessing Software Measurement Programs	Michael Berry, Ross Jeffery	Empirical Software Engineering, Volume 5, Issue 3, Nov 2000, Pages 183 - 200
55	An Instrument for Measuring the Key Factors of Success in Software Process Improvement	Tore Dybå	Empirical Software Engineering, Volume 5, Issue 4, Dec 2000, Pages 357 - 390
56	Annual Index	IEEE	Software, IEEE , vol.21, no.6pp. 95- 99, Nov.-Dec. 2004
57	Applying and adjusting a software process improvement model in practice: the use of the IDEAL model in a small software enterprise	Karlheinz Kautz, Henrik Westergaard Hansen, Kim Thaysen	Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, Pages: 626 - 633, 2000
58	Applying Benchmarking to Learn from Best Practices	Andrew Beitz, Isabella Wieczorek	Lecture Notes in Computer Science, Volume 1840, Jan 2000, Pages 59 - 72
59	Applying software process modeling and improvement in academic setting	Jaccheri, M.L.; Lago, P.	Software Engineering Education & Training. Tenth Conference on , vol., no.pp.13-27, 1-16 Apr 1997
60	Applying SPICE to e-learning: an e-learning maturity model?	Stephen Marshall, Geoff Mitchell	Proceedings of the sixth conference on Australian computing education - Volume 30, Dunedin, New Zealand, Pages: 185 - 191, 2004
61	Assessing attitude towards, knowledge of, and ability to apply, software development process	Klappholz, D.; Bernstein, L.; Port, D.	Software Engineering Education and Training, 2003. (CSEE&T 2003). Proceedings. 16th Conference on , vol., no.pp. 268- 278, 20-22 March 2003
62	Assessing the Changeability of two Object-Oriented Design Alternatives--a Controlled Experiment	Erik Arisholm, Dag I. K. Sjøberg, Magne Jørgensen	Empirical Software Engineering, Volume 6, Issue 3, Sep 2001, Pages 231 - 277
63	Assessing usability through perceptions of information scent	Saward, G.; Hall, T.; Barker, T.,	Software Metrics, 2004. Proceedings. 10th International Symposium on , vol., no.pp. 337- 346, 14-16 Sept. 2004
64	Assessment of CAC self-study report	Cook, C.; Mathur, P.; Visconti, M.	Frontiers in Education, 2004. FIE 2004. 34th Annual , vol., no.pp. T3G/12- T3G/17 Vol. 1, 20-23 Oct. 2004
65	Assessment of software measurement: an information quality study	Berry, M.; Jeffery, R.; Aurum, A.	Software Metrics, 2004. Proceedings. 10th International Symposium on , vol., no.pp. 314- 325, 14-16 Sept. 2004
66	Assessments of usability engineering processes: experiences from experiments	Jokela, T.,	System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on , vol., no.pp. 9 pp.-, 6-9 Jan. 2003
67	Assuring fault classification agreement - an empirical evaluation	Henningsson, K.; Wohlin, C.	Empirical Software Engineering, 2004. ISESE '04. Proceedings. 2004 International Symposium on , vol., no.pp. 95- 104, 19-20 Aug. 2004
68	Benchmarking software organizations	Card, D.; Zubrow, D.	Software, IEEE , vol.18, no.5pp.16-17, Sep/Oct 2001
69	Blending CMM and Six Sigma to meet business goals	Murugappan, M.; Keeni, G.	Software, IEEE , vol.20, no.2pp. 42- 48, Mar/Apr 2003
70	Capability Maturity Model software development using Cleanroom software engineering principles-results of an industry project	Oshana, R.S.; Linger, R.C.,	System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on , vol.Track7, no.pp.10 pp.-, 1999
71	Capability maturity model, version 1.1	Paulk, M.C.; Curtis, B.; Chrissis, M.B.; Weber, C.V.	Software, IEEE , vol.10, no.4pp.18-27, Jul 1993
72	Capturing evidence from wireless Internet services development	Bella, F.; Munch, J.; Ocampo, A.	Software Technology and Engineering Practice, 2003. Eleventh Annual International Workshop on , vol., no.pp. 33- 39, 19-21 Sept. 2003
73	Case studies of software-process-improvement measurement	Paulish, D.J.; Carleton, A.D.	Computer , vol.27, no.9pp.50-57, Sep 1994
74	Certification and licensing for software professionals and organizations	Werth, L.	Software Engineering Education, 1998. Proceedings., 11th Conference on , vol., no.pp.151-160, 22-25 Feb 1998
75	Characterizing a software process maturity model for small organizations	Hossein Saiedian, Natsu Carr	ACM SIGICE Bulletin, Volume 23 , Issue 1 (July 1997), Pages: 2 - 11, 1997
76	Cognizant enterprise maturity model (CEMM)	Harigopal, U.; Satyadas, A.,	Systems, Man and Cybernetics, Part C, IEEE Transactions on , vol.31, no.4pp.449-459, Nov 2001
77	Collaborations: closing the industry-academia gap	Beckman, K.; Coulter, N.; Khajenoori, S.; Mead, N.R.	Software, IEEE , vol.14, no.6pp.49-57, Nov/Dec 1997
78	Collecting Feedback during Software Engineering Experiments	Amela Karahasanoviae, Bente Anda, Erik Arisholm, Siw Elisabeth Hove, Magne Jørgensen, Dag I K Sjøberg, Ray Welland	Empirical Software Engineering, Volume 10, Issue 2, Apr 2005, Pages 113 - 147
79	Combining agile methods with stage-gate project management	Karlstrom, D.; Runeson, P.	Software, IEEE , vol.22, no.3pp. 43- 49, May-June 2005
80	Combining quality and software improvement	Craig Hollenbach, Ralph Young, Al Pflugrad, Doug Smith	Communications of the ACM, Volume 40 , Issue 6 (June 1997), Pages: 41 - 45, 1997
81	Comments on 'a critical look' [software capability evaluations]	Humphrey, W.S.; Curtis, B.	Software, IEEE , vol.8, no.4pp.42-46, Jul 1991
82	Commitment development in software process improvement: critical misconceptions	Pekka Abrahamsson	Proceedings of the 23rd International Conference on Software Engineering, Toronto, Ontario, Canada, Pages: 71 - 80, 2001

83	Commitment in software process improvement - in search of the process	Abrahamsson, P.; Iivari, N.	System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on , vol., no.pp. 3239- 3248, 7-10 Jan. 2002
84	Commitment Nets in Software Process Improvement	Pekka Abrahamsson	Annals of Software Engineering, Volume 14, Issue 1 - 4, Dec 2002, Pages 407 - 438
85	Competing values in software process improvement: an assumption analysis of CMM from an organizational culture perspective	Ngwenyama, O.; Nielsen, P.A.	Engineering Management, IEEE Transactions on , vol.50, no.1pp. 100- 112, Feb. 2003
86	Conducting realistic experiments in software engineering	Sjoberg, D.I.K.; Anda, B.; Arisholm, E.; Dyba, T.; Jorgensen, M.; Karahasanovic, A.; Koren, E.F.; Vokac, M.	Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n , vol., no.pp. 17- 26, 2002
87	Contextualizing Knowledge Management Readiness to Support Change Management Strategies	Keith, M.; Goul, M.; Demirkan, H.; Nichols, J.; Mitchell, M.C.	System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on , vol.7, no.pp. 152a- 152a, 04-07 Jan. 2006
88	Corporate split couldn't keep Lucent's IS team from pulling together employees	Ilan Greenberg	InfoWorld. San Mateo: Sep 29, 1997. Vol. 19, Num. 39; pág. 96 (1 página)
89	Creating a Dual-Agility Method: The Value of Method Engineering	B Henderson-Sellers, M K Serour	Journal of Database Management, Hershey: Oct-Dec 2005.Vol.16, Num. 4; pg. 1, 23 pgs
90	Cultural Impact on Intergroup Coordination in Software Development in China: A Qualitative Analysis	Minghui Yuan; Vogel, D.	System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on , vol.1, no.pp. 13c- 13c, 04-07 Jan. 2006
91	Decision support for extreme programming introduction and practice selection	Daniel Karlström, Per Runeson	Proceedings of the 14th international conference on Software engineering and knowledge engineering, Ischia, Italy, Pages: 835 - 841, 2002
92	Defect prevention through defect prediction: a case study at Infosys	Mohapatra, S.; Mohanty, B.	Software Maintenance, 2001. Proceedings. IEEE International Conference on , vol., no.pp.260-272, 2001
93	Designing and Conducting an Empirical Study on Test Management Automation	Griselda Giraudo, Paolo Tonella	Empirical Software Engineering, Volume 8, Issue 1, Mar 2003, Pages 59 - 81
94	Determining Inspection Cost-Effectiveness by Combining Project Data and Expert Opinion	Freimut, B.; Briand, L.C.; Vollei, F.	Software Engineering, IEEE Transactions on , vol.31, no.12pp. 1074- 1092, Dec. 2005
95	Developing a common operating environment for military application	Jungyoon Kim; Joon-Sang Lee; Doo Hwan Bae; Dong-Kuk Ryu; Sang-Il Lee	Distributed Computing Systems, 2003. FTDCS 2003. Proceedings. The Ninth IEEE Workshop on Future Trends of , vol., no.pp. 367- 373, 28-30 May 2003
96	Developing organizational competence	Curtis, B.; Hefley, W.E.; Miller, S.; Konrad, M.	Computer , vol.30, no.3pp.122-124, Mar 1997
97	Developing, validating and evolving an approach to product line benefit and risk assessment	Schmid, K.; John, I.	Euromicro Conference, 2002. Proceedings. 28th , vol., no.pp. 272- 283, 2002
98	Development process assessment toward leading edge quality	Botten, N.A.	Selected Areas in Communications, IEEE Journal on , vol.12, no.2pp.251-257, Feb 1994
99	Difficulties in Managing Offshore Software Outsourcing Relationships: An Empirical Analysis of 18 High Maturity Indian Software Companies	Nilay V Oza, Tracy Hall.	Journal of Information Technology Case and Application Research. Marietta: 2005.Vol.7, Num. 3; pg. 25, 17 pgs
100	Diffusion of software technology innovations in the global context	Nahar, N.; Kakola, T.; Huda, N.	System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on , vol., no.pp. 2749- 2757, 7-10 Jan. 2002
101	Does Use of Development Model Affect Estimation Accuracy and Bias?	Kjetil Møløkken, Anette C. Lien, Magne Jørgensen, Sinan S.	Lecture Notes in Computer Science, Volume 3009, Jan 2004, Pages 17 - 29
102	Effective test metrics for test strategy evolution	Yanping Chen, Robert L. Probert, Kyle Robeson	Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research, Markham, Ontario, Canada, Pages: 111 - 123, 2004
103	Elements of a realistic CASE tool adoption budget	Clifford C. Huff	Communications of the ACM, Volume 35 , Issue 4 (April 1992), Pages: 45 - 54, 1992
104	Embedded values in process and practice: Interactions between disciplinary practices and formal innovation processes	Peter H Jones	Design Management Journal. 2002. Vol. 2; pág. 20
105	Empirical Studies Applied to Software Process Models	David Raffo, Timo Kaitio, Derek Partridge, Keith Phalp, Juan F. Ramil	Empirical Software Engineering, Volume 4, Issue 4, Dec 1999, Pages 353 - 369
106	Empirically Driven Design of Software Development Processes for Wireless Internet Services	Ulrike Becker-Kornstaedt, Daniela Boggio, Jürgen Münch, Alexis Ocampo, Gino Palladino	Lecture Notes in Computer Science, Volume 2559, Jan 2002, Pages 351 - 366
107	Enabling Software Process Improvement: An Investigation of the Importance of Organizational Issues	Tore Dybå	Empirical Software Engineering, Volume 7, Issue 4, Dec 2002, Pages 387 - 390
108	Enhanced software project management by application of metrics and cost estimation techniques	Fitzhenry, P.; Gardiner, G.,	Project Management for Software Engineers, IEE Colloquium on , vol., no.pp.4/1-4/4, 11 Dec 1995
109	Establishing experience factories at Daimler-Benz: an experience report	Frank Houdek, Kurt Schneider, Eva Wieser	Proceedings of the 20th international conference on Software engineering, Kyoto, Japan, Pages: 443 - 447, 1998

110	Estimating Software Development Effort Based on Use Cases- Experiences from Industry	Bente Anda, Hege Dreiem, Dag I.K. Sjøberg, Magne Jørgensen	Lecture Notes in Computer Science, Volume 2185, Jan 2001, Page 487
111	European experiences with software process improvement	Fran O'Hara	Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, Pages: 635 - 640, 2000
112	Evaluating a Rapid Simulation Modelling Process (RSMP) through Controlled Experiments	Ahmed, R.; Hall, T.; Wernick, P.; Robinson, S.	Empirical Software Engineering, 2005. 2005 International Symposium on , vol., no.pp. 314- 323, Nov. 17, 2005
113	Evaluating the effect of a delegated versus centralized control style on the maintainability of object-oriented software	Arisholm, E.; Sjøberg, D.I.K.	Software Engineering, IEEE Transactions on , vol.30, no.8pp. 521- 534, Aug. 2004
114	Evaluation of a metrics framework for product and process integrity	McKeown, K.A.; McGuire, E.G.	System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on , vol., no.pp. 6 pp.-, 4-7 Jan. 2000
115	Evaluation of code review methods through interviews and experimentation	Martin Höst and Conny Johansson	Journal of Systems and Software, Volume 52, Issues 2-3, 1 June 2000, Pages 113-120
116	Evidence-based software engineering for practitioners	Dyba, T.; Kitchenham, B.A.; Jorgensen, M.	Software, IEEE , vol.22, no.1pp. 58- 65, Jan.-Feb. 2005
117	Evolution of the frameworks quagmire	Sheard, S.A.	Computer , vol.34, no.7pp.96-98, Jul 2001
118	Experience in implementing a learning software organization	Schneider, K.; von Hunnius, J.-P.; Basili, V.R.	Software, IEEE , vol.19, no.3pp.46-49, May/June 2002
119	Experience with software process in physics projects	Guatelli, S.; Mascialino, B.; Moneta, L.; Papadopoulos, I.; Pfeiffer, A.; Pia, M.G.; Piergentili, M.	Nuclear Science Symposium Conference Record, 2004 IEEE , vol.4, no.pp. 2100- 2103 Vol. 4, 16-22 Oct. 2004
120	Experience-based model-driven improvement management with combined data sources from industry and academia	Jedlitschka, A.; Pfahl, D.	Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on , vol., no.pp. 154- 161, 30 Sept.-1 Oct. 2003
121	Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research	Hove, S.E.; Anda, B.	Software Metrics, 2005. 11th IEEE International Symposium , vol., no.pp. 23- 23, 19-22 Sept. 2005
122	Experiences from the pilot operation and commissioning phase of a SCM process improvement program	Nattinen, M.; Rahikkala, T.; Valimaki, A.,	EUROMICRO Conference, 1999. Proceedings. 25th , vol.2, no.pp.185-192 vol.2, 1999
123	Experiences in the Application of Software Process Improvement in SMES	Jose A. Calvo-Manzano Villalón, Gonzalo Cuevas Agustín, Tomás San Feliu Gilabert, Antonio De Amescua Seco, et al.	Lecture Notes in Computer Science, Volume 3547, Jun 2005, Pages 216 - 223
124	Experiences on defining and evaluating an adapted review process	Thomas Gantner, Tobias Barth	Proceedings of the 25th International Conference on Software Engineering, Portland, Oregon, Pages: 506 - 511, 2003
125	Explaining software developer acceptance of methodologies: a comparison of five theoretical models	Riemenschneider, C.K.; Hardgrave, B.C.; Davis, F.D.	Software Engineering, IEEE Transactions on , vol.28, no.12pp. 1135- 1145, Dec 2002
126	Exploring bottlenecks in market-driven requirements management processes with discrete event simulation	Martin Höst, Björn Regnell, Johan Natt och Dag, Josef Nedstam and Christian Nyberg	Journal of Systems and Software, Volume 59, Issue 3, 15 December 2001, Pages 323-332
127	Exploring extreme programming in context: an industrial case study	Layman, L.; Williams, L.; Cunningham, L.	Agile Development Conference, 2004 , vol., no.pp. 32- 41, 22-26 June 2004
128	Exploring the Context of Product Line Adoption	Stan Bühne, Gary Chastek, Timo Kähkölä, Peter Knauber, Linda Northrop, Steffen Thiel	Lecture Notes in Computer Science, Volume 3014, Jan 2004, Pages 19 - 31
129	Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context	Tore Dybå	Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering, Helsinki, Finland, Pages: 148 - 157,
130	From MCC to CMM: technology transfers bright and dim	Bill Curtis	Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, Pages: 521 - 530, 2000
131	From the many to the one-one company's path to standardization	Cromer, T.; Horch, J.,	Software Engineering Standards, 1999. Proceedings. Fourth IEEE International Symposium and Forum on , vol., no.pp.116-121, 1999
132	Frontmatter (letters and notices)	Will Tracz	ACM SIGSOFT Software Engineering Notes, Volume 29 , Issue 3 (May 2004), Pages: 0 - 4, 2004
133	Frontmatter (TOC, Letters, Election results, Software Reliability Resources!, Computing Curricula 2004 and the Software Engineering Volume SE2004, Software Reuse Research, ICSE 2005 Forward)	ACM	ACM SIGSOFT Software Engineering Notes, Volume 30 , Issue 4 (July 2005), 2005
134	Future Studies of Learning Software Organizations	Kari Smolander, Kurt Schneider, Torgeir Dingsøy, Finn Olav Bjørnsson, Pasi Juvonen, Päivi Ovaska	Lecture Notes in Computer Science, Volume 3782, Dec 2005, Pages 134 - 144

135	Generalised environment for process management in cooperative software engineering	Gaeta, M.; Ritrovato, P.	Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International , vol., no.pp. 1049- 1053, 2002
136	Gray rebuts Bach: no cowboy programmers!	Gray, L.	Computer , vol.31, no.4pp.102-103, 105, Apr 1998
137	How to use linguistic instruments for object-oriented analysis	Juristo, N.; Moreno, A.M.; Lopez, M.,	Software, IEEE , vol.17, no.3pp.80-89, May/Jun 2000
138	Human and social factors of software engineering: workshop summary	Michael John, Frank Maurer, Bjørnar Tessem	ACM SIGSOFT Software Engineering Notes, Volume 30 , Issue 4 (July 2005), Pages: 1 - 6, 2005
139	Identification of key factors in software process management - a case study	Berander, P.; Wohlin, C.	Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on , vol., no.pp. 316- 325, 30 Sept.-1 Oct. 2003
140	Identifying extensions required by RUP (rational unified process) to comply with CMM (capability maturity model) levels 2 and 3	Manzoni, L.V.; Price, R.T.	Software Engineering, IEEE Transactions on , vol.29, no.2pp. 181- 192, Feb. 2003
141	IEEE Std 1348-1995. IEEE recommended practice for the adoption of Computer-Aided Software Engineering (CASE) tools	IEEE	IEEE Std 1348-1995 , vol., no.pp.-, 10 Apr 1996
142	Implementing requirements engineering processes throughout organizations: success factors and challenges	Marjo Kauppinen, Matti Vartiainen, Jyrki Kontio, Sari Kujala and Reijo Sulonen	Information and Software Technology, Volume 46, Issue 14, 1 November 2004, Pages 937-953
143	Implications of evolution metrics on software maintenance	Lehman, M.M.; Perry, D.E.; Ramil, J.F.	Software Maintenance, 1998. Proceedings. International Conference on , vol., no.pp.208-217, 16-20 Nov 1998
144	Improving diffusion practices in a software organization	Andersson, I.; Nilsson, K.	System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on , vol., no.pp. 8 pp.-, 7-10 Jan. 2002
145	Improving Dynamic Calibration through Statistical Process Control	Baldassarre, M.T.; Boffoli, N.; Caivano, D.; Visaggio, G.	Software Maintenance, 2005. ICSM'05. Proceedings of the 21st IEEE International Conference on , vol., no.pp. 273- 282, 2005
146	Improving hypermedia development: a reference model-based process assessment method	David B. Lowe, Andrew J. Bucknell, Richard G. Webby	Proceedings of the tenth ACM Conference on Hypertext and hypermedia : returning to our diverse roots: returning to our diverse roots, Darmstadt, Germany, Pages: 139 - 146, 1999
147	Improving Validation Activities in a Global Software Development	Christof Ebert, Casimiro Hernandez Parro, Roland Suttels, Harald Kolarczyk	Proceedings of the 23rd International Conference on Software Engineering, Toronto, Ontario, Canada, Pages: 545 - 554, 2001
148	Increasing understanding of the modern testing perspective in software development projects	Pyhajarvi, M.; Rautiainen, K.; Itkonen, J.	System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on , vol., no.pp. 10 pp.-, 6-9 Jan. 2003
149	Industrial strength software and quality: software and engineering at Siemens	Achatz, R.; Paulisch, F.,	Quality Software, 2003. Proceedings. Third International Conference on , vol., no.pp. 321- 326, 6-7 Nov. 2003
150	Integrating business and software development models	Wallin, C.; Ekdahl, F.; Larsson, S.	Software, IEEE , vol.19, no.6pp. 28- 33, Nov/Dec 2002
151	Integrating forward and reverse object-oriented software engineering	Christoph Welsch, Alexander Schalk, Stefan Kramer	Proceedings of the 19th international conference on Software engineering, Boston, Massachusetts, United States, Pages: 560 - 561, 1997
152	Introducing a software modeling concept in a medium-sized company	Schmid, K.; Becker-Kornstaedt, U.; Kanuber, P.; Bernauer, F.,	Software Engineering, 2000. Proceedings of the 2000 International Conference on , vol., no.pp.558-567, 2000
153	Introducing requirements engineering: how to make a cultural change happen in practice	Kauppinen, M.; Kujala, S.; Aalio, T.; Lehtola, L.	Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on , vol., no.pp. 43-51, 2002
154	Introducing workflow management in software maintenance processes	Aversano, L.; Betti, S.; De Lucia, A.; Stefanucci, S.	Software Maintenance, 2001. Proceedings. IEEE International Conference on , vol., no.pp.441-450, 2001
155	Introduction of modern software technology. A case from Swedish industry	J. S. Jaliff, C. Å. Jonsson, A. Ljungblad	Software Quality Journal, Volume 6, Issue 1, Mar 1997, Pages 1 - 11
156	Investigating the accuracy of defect estimation models for individuals and teams based on inspection data	Biffi, S.; Halling, M.; Koszegi, S.,	Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on , vol., no.pp. 232- 243, 30 Sept.-1 Oct. 2003
157	Investigating the Role of Use Cases in the Construction of Class Diagrams	Bente Anda, Dag I. K. Sjøberg	Empirical Software Engineering, Volume 10, Issue 3, Jul 2005, Pages 285 - 309
158	Investigating training effects on software reviews: a controlled experiment	Wee Land, L.P.; Tan, B.C.Y.; Bin, L.	Empirical Software Engineering, 2005. 2005 International Symposium on , vol., no.pp. 344- 354, Nov. 17, 2005
159	Is a Design Rationale Vital when Predicting Change Impact? - A Controlled Experiment on Software Architecture Evolution	Lars Bratthall , Enrico Johansson, and Björn Regnell	Lecture Notes in Computer Science, Volume 1840, Jan 2000, Pages 126 - 139
160	Is Extreme Programming Just Old Wine in New Bottles: A Comparison of Two Cases	Merisalo-Rantanen Hilikka, Tuunanen Tuure, Rossi Matti	Journal of Database Management. Hershey: Oct-Dec 2005.Vol.16, Num. 4; pg. 41, 21 pgs
161	Is the software engineering state of the practice getting closer to the of the art?	Reifer, D.J.	Software, IEEE , vol.20, no.6pp. 78- 83, Nov.-Dec. 2003
162	Issues in using students in empirical studies in software engineering education	Carver, J.; Jaccheri, L.; Morasca, S.; Shull, F.	Software Metrics Symposium, 2003. Proceedings. Ninth International , vol., no.pp. 239- 249, 3-5 Sept. 2003
163	Key lessons in achieving widespread inspection use	Grady, R.B.; Slack, T.V.	Software, IEEE , vol.11, no.4pp.46-57, Jul 1994

164	Knowledge management in software engineering - describing the process	Ward, J.; Aurum, A.	Software Engineering Conference, 2004. Proceedings. 2004 Australian , vol., no.pp. 137- 146, 2004
165	Knowledge Support in Software Process Tailoring	Peng Xu	System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on , vol., no.pp. 87c- 87c, 03-06 Jan. 2005
166	Knowledge-Sharing Issues in Experimental Software Engineering	Forrest Shull, Manoel G. Mendonça, Victor Basili, Jeffrey Carver, José C. Maldonado, Sandra Fabbri, Guilherme Horta Travassos, Maria Cristina Ferreira	Empirical Software Engineering, Volume 9, Issue 1 - 2, Mar 2004, Pages 111 - 137
167	Learning dysfunctions in information systems development: Overcoming the social defenses with transitional objects	David G Wastell	MIS Quarterly. Minneapolis: Dec 1999. Vol. 23, Num. 4; pág. 581 (20 páginas)
168	Learning from success [business process improvement]	Nolan, A.J.,	Software, IEEE , vol.16, no.1pp.97-105, Jan/Feb 1999
169	Lessons Learned and Recommendations from Two Large Norwegian SPI Programmes	Reidar Conradi Tore Dybå1, Dag I.K. Sjøberg and Tor Ulsund	Lecture Notes in Computer Science, Volume 2786, Jan 2003, Pages 32 - 45
170	Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory	Victor R. Basili, Frank E. McGarry, Rose Pajerski, Marvin V. Zelkowitz	Proceedings of the 24th International Conference on Software Engineering, Orlando, Florida, Pages: 69 - 79, 2002
171	Lessons learned implementing ISO 9001 in a software organization	Ganner, M.; Johnson, M.	Innovation in Technology Management - The Key to Global Leadership. PICMET '97: Portland International Conference on Management and Technology , vol., no.pp.606-608, 27-31 Jul 1997
172	Lessons learned in framework-based software process improvement	Jalote, P.	Software Engineering Conference, 2002. Ninth Asia-Pacific , vol., no.pp. 261- 265, 2002
173	Limits in modelling evolving computer-based systems	Massimo Felici, Juliana Küster Filipe	Proceedings of the 2002 ACM symposium on Applied computing, Madrid, Spain, Pages: 706 - 710, 2002
174	Making sense of measurement for small organizations	Kautz, K.	Software, IEEE , vol.16, no.2pp.14-20, Mar/Apr 1999
175	Making SPI happen: the IDEAL distribution of effort	Borjesson, A.; Mathiassen, L.	System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on , vol., no.pp. 10 pp.-, 6-9 Jan. 2003
176	Managing process diversity while improving your practices	Beck, M.	Software, IEEE , vol.18, no.3pp.21-27, May 2001
177	Managing Risk In Software Process Improvement: An Action Research Approach	Jakob H Iversen, Lars Mathiassen, Peter Axel Nielsen	MIS Quarterly. Minneapolis: Sep 2004. Vol. 28, Num. 3; pg. 395, 39 pgs
178	Managing software productivity and reuse	Boehm, B.,	Computer , vol.32, no.9pp.111-113, Sep 1999
179	Measurement processes are software, too	M. Morisio	Journal of Systems and Software, Volume 49, Issue 1, 15 December 1999, Pages 17-31
180	Measurement program success factors revisited	Frank Niessink and Hans van Vliet	Information and Software Technology, Volume 43, Issue 10, 15 August 2001, Pages 617-628
181	Measurement programs in software development: determinants of success	Gopal, A.; Krishnan, M.S.; Mukhopadhyay, T.; Goldenson, D.R.	Software Engineering, IEEE Transactions on , vol.28, no.9pp. 863- 875, Sep 2002
182	Measurement: The key to application development quality	Walrad, Charlene, Moss, Eric	IBM Systems Journal. Armonk: 1993. Vol. 32, Num. 3; pág. 445 (16 páginas)
183	Measuring process consistency: implications for reducing software defects	Krishnan, M.S.; Kellner, M.I.	Software Engineering, IEEE Transactions on , vol.25, no.6pp.800-815, Nov/Dec 1999
184	Measuring software quality: a case study	Drake, T.	Computer , vol.29, no.11pp.78-87, Nov 1996
185	Metrics for meta-heuristic algorithm evaluation	Qian-Li Zhang,	Machine Learning and Cybernetics, 2003 International Conference on , vol.2, no.pp. 1241-1244 Vol.2, 2-5 Nov. 2003
186	Metrics for small projects: Experiences at the SED	Grable, R.; Jernigan, J.; Pogue, C.; Divis, D.,	Software, IEEE , vol.16, no.2pp.21-29, Mar/Apr 1999
187	Models for the evolution of OS projects	Capiluppi, A.,	Software Maintenance, 2003. ICSM 2003. Proceedings. International Conference on , vol., no.pp. 65- 74, 22-26 Sept. 2003
188	Multi-method research: An empirical investigation of object-oriented technology	Murray Wood, John Daly, James Miller and Marc Roper	Journal of Systems and Software, Volume 48, Issue 1, 1 August 1999, Pages 13-26
189	On the limitations of software process assessment and the recognition of a required re-orientation for global process improvement	E.M. Gray, W.L. Smith	Software Quality Journal, Volume 7, Issue 1, Mar 1998, Pages 21 - 34
190	One XP experience: introducing agile (XP) software development into a culture that is willing but not ready	F. Grossman, J. Bergin, D. Leip, S. Merritt, O. Gotel	Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research, Markham, Ontario, Canada, Pages: 242 - 254, 2004
191	Organizational change as a contributing factor to IS failure	Winklhofer, H.	System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on , vol., no.pp. 9 pp.-, 3-6 Jan. 2001
192	Papers by Author	IEEE	System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on , vol., no.pp. 0. 225- 0. 444, 3-6 Jan. 2001

193	Perceptions of software quality: a pilot study	David N. Wilson, Tracy Hall	Software Quality Journal, Volume 7, Issue 1, Mar 1998, Pages 67 - 75
194	Principles of survey research: part 3: constructing a survey instrument	Barbara A. Kitchenham, Shari Lawrence Pfleeger	ACM SIGSOFT Software Engineering Notes, Volume 27, Issue 2 (March 2002), COLUMN: Columns table of contents, Pages: 20 - 24, 2002
195	PRISM: a systematic approach to planning technology transfer campaigns	Capps, B.; Fairley, R.E.	Management of Engineering and Technology, 2003. PICMET '03. Technology Management for Reshaping the World. Portland International Conference on, vol., no.pp. 393- 399, 20-24 July 2003
196	Proceedings 27th Annual International Computer Software and Applications Conference. COMPAC 2003	COMPAC 2003	Computer Software and Applications Conference, 2003. COMPSAC 2003. Proceedings. 27th Annual International, vol., no.pp.-, 3-6 Nov. 2003
197	Proceedings of the 33rd Annual Hawaii International Conference on System Sciences	IEEE	System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on, vol., no.pp.-, 4-7 Jan. 2000
198	Proceedings of the 34th Annual Hawaii International Conference on System Sciences	IEEE	System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on, vol., no.pp.-, 3-6 Jan. 2001
199	Process improvement and the corporate balance sheet	Dion, R.	Software, IEEE, vol.10, no.4pp.28-35, Jul 1993
200	Process improvement competencies for IS professionals: a survey of perceived needs	Eugene G. McGuire, Kim A. Randall	Proceedings of the 1998 ACM SIGCPR conference on Computer personnel research, Boston, Massachusetts, United States, Pages: 1 - 8, 1998
201	Process improvement for small firms: An evaluation of the RAPID assessment-based method	Aileen Cater-Steel, Mark Toleman and Terry Rout	Information and Software Technology, In Press, Corrected Proof, Available online 20 December 2005
202	Process improvement for small organizations	Kelly, D.P.; Culleton, B.	Computer, vol.32, no.10pp.41-47, Oct 1999
203	Process improvement in facilities management: the SPICE approach	Dilanthe Amaratunga, Marjan Sarshar, David Baldry	Business Process Management Journal. Bradford: 2002. Vol. 8, Num. 4; pág. 318 (20 páginas)
204	Process improvement towards ISO 9001 certification in a small software organization	Elif Demirörs, Onur Demirörs, Oguz Dikenelli, Billur Keskin	Proceedings of the 20th international conference on Software engineering, Kyoto, Japan, Pages: 435 - 438, 1998
205	Process Management in Practice Applying the FUNSOFT Net Approach to Large-Scale Processes	Wolfgang Deiters, Volker Gruhn	Automated Software Engineering, Volume 5, Issue 1, Jan 1998, Page 7
206	Product Focused Software Process Improvement: Concepts and Experiences from Industry	Jos Trienekens, Rob Kusters, Rini Van Solingen	Software Quality Journal, Volume 9, Issue 4, Dec 2001, Pages 269 - 281
207	Progressive open source	Jamie Dinkelacker, Pankaj K. Garg, Rob Miller, Dean Nelson	Proceedings of the 24th International Conference on Software Engineering, Orlando, Florida, Pages: 177 - 184, 2002
208	Putting the Personal Software ProcessSM into practice	Cannon, R.L.	Software Engineering Education and Training, 1999. Proceedings. 12th Conference on, vol., no.pp.34-37, 22-24 Mar 1999
209	Quality management in systems development: An organizational system perspective	T Ravichandran, Arun Rai	MIS Quarterly. Minneapolis: Sep 2000. Vol. 24, Num. 3; pág. 381 (35 páginas)
210	Quality through Managed Improvement and Measurement (QMIM): Towards a Phased Development and Implementation of a Quality Management System for a Software Company	"Katalin Balla, Theo Bemelmans, Rob Kusters, Jos Trienekens	Software Quality Journal, Volume 9, Issue 3, Nov 2001, Pages 177 - 193
211	Raytheon lays down a practices path for developers to follow	Strehlo, Kevin	InfoWorld. San Mateo: Jun 24, 1996. Vol. 18, Num. 26; pág. 85 (1 página)
212	Reinventing methodology: who reads it and why?	Gezinus J. Hidding	Communications of the ACM, Volume 40, Issue 11 (November 1997), Pages: 102 - 109, 1997
213	Replicating software engineering experiments: addressing the tacit knowledge problem	Shull, F.; Basili, V.; Carver, J.; Maldonado, J.C.; Travassos, G.H.; Mendonca, M.; Fabbri, S.	Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium on, vol., no.pp. 7- 16, 2002
214	Requirements engineering process improvement based on an information model	Doerr, J.; Paech, B.; Koehler, M.,	Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International, vol., no.pp. 70- 79, 6-11 Sept. 2004
215	Results from introducing component-level test automation and Test-Driven Development	Lars-Ola Damm and Lars Lundberg	Journal of Systems and Software, In Press, Corrected Proof, Available online 5 December 2005
216	Results of applying the Personal Software Process	Ferguson, P.; Humphrey, W.S.; Khajenoori, S.; Macke, S.; Matvya, A.	Computer, vol.30, no.5pp.24-31, May 1997
217	Reusable software components (1996)	Trudy Levine	ACM SIGAda Ada Letters, Volume XVI, Issue 1 Jan./Feb. 1996, Pages: 25 - 35, 1996
218	Reusable software components (1998)	Trudy Levine	ACM SIGAda Ada Letters, Volume XVIII, Issue 4 July/Aug. 1998, Pages: 32 - 46, 1998
219	Reuse in Command-and-Control Systems	O'Connor, J.; Mansour, C.; Turner-Harris, J.; Campbell, G.H., Jr.	Software, IEEE, vol.11, no.5pp.70-79, Sep 1994
220	Second Workshop on Software Quality	Bernard Wong, Sunita Chulani, June Verner, Barry Boehm	Proceedings of the 26th International Conference on Software Engineering, Pages: 780 - 782, 2004

221	Siemens process assessment and improvement approaches: experiences and benefits	Mehner, T.; Messer, T.; Paul, P.; Paulisch, F.; Schless, P.; Volker, A.	Computer Software and Applications Conference, 1998. COMPSAC '98. Proceedings. The Twenty-Second Annual International , vol., no.pp.186-195, 19-21 Aug 1998
222	Silver Pellets for Improving Software Quality	Evan W Duggan	Information Resources Management Journal. Hershey: Apr-Jun 2004.Vol.17, Num. 2; pg. 1, 21 pgs
223	Situated assessment of problems in software development	Jakob Iversen, Peter Axel Nielsen, Jacob Norbjerg	Database for Advances in Information Systems. New York: Spring 1999. Vol. 30, Num. 2; pág. 66 (16 páginas)
224	Six Sigma in the software industry: results from a pilot study	Jiju Antony, Craig Fergusson	Managerial Auditing Journal. Bradford: 2004. Vol. 19, Num. 8/9; pg. 1025
225	SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)	IEEE	Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on , vol.1, no.pp.-, 11-14 Oct. 1998
226	Software and business process technology (tutorial)	Volker Gruhn, Wilhelm Schäfer	Proceedings of the 19th international conference on Software engineering, Boston, Massachusetts, United States, Pages: 639 - 640, 1997
227	Software Development and Experimentation in an Academic Environment: The Gaudi Experience	Ralph-Johan Back, Luka Milovanov, Ivan Porres	Lecture Notes in Computer Science, Volume 3547, Jun 2005, Pages 414 - 428
228	Software development method tailoring at Motorola	Brian Fitzgerald, Nancy L. Russo, Tom O'Kane	Communications of the ACM, Volume 46 , Issue 4 (April 2003), Digital rights management, Pages: 64 - 70, 2003
229	Software Development Process Improvement: The Forgotten Son?	Gregg Tong	World Class Design to Manufacture; Volume: 1 Issue: 5; 1994
230	Software development: Processes and performance	S Sawyer, P J Guinan	IBM Systems Journal
231	Software engineering for semiconductor manufacturing equipment suppliers	Baudoin, C.R.; Kantor, J.P.	Components, Packaging, and Manufacturing Technology, Part A, IEEE Transactions on [see also Components, Hybrids, and Manufacturing Technology, IEEE Transactions on] , vol.17, no.2pp.230-243, Jun 1994
232	Software engineering standards and the development of multimedia-based systems	Rout, T.P.; Sherwood, C.	Software Engineering Standards, 1999. Proceedings. Fourth IEEE International Symposium and Forum on , vol., no.pp.192-198, 1999
233	Software engineering technology infusion within NASA	Zelkowitz, M.V.	Engineering Management, IEEE Transactions on , vol.43, no.3pp.250-261, Aug 1996
234	Software Experience Bases: A Consolidated Evaluation and Status Report	Reidar Conradi, Torgeir Dingsøyr	Lecture Notes in Computer Science, Volume 1840, Jan 2000, Pages 391 - 406
235	Software process improvement as organizational knowledge creation: a multiple case analysis	Arent, J.; Norbjerg, J.	System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on , vol., no.pp. 11 pp.-, 4-7 Jan. 2000
236	Software process improvement experience in the DP/MIS function: experience report	Albert Johnson	Proceedings of the 16th international conference on Software engineering, Sorrento, Italy, Pages: 323 - 329, 1994
237	Software process improvement in an automotive electronics organization	McHugh, G.C.	Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE , vol.2, no.pp.111/1-111/6 vol.2, 31 Oct-7 Nov 1998
238	Software process improvement in Web time	Wiegiers, K.E.	Software, IEEE , vol.16, no.4pp.78-86, Jul/Aug 1999
239	Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis	Sarah Beecham, Tracy Hall, Austen Rainer	Empirical Software Engineering, Volume 8, Issue 1, Mar 2003, Page 7
240	Software process improvement: Blueprints versus recipes	Aaen, I.	Software, IEEE , vol.20, no.5pp. 86- 93, Sept.-Oct. 2003
241	Software process improvement: making it happen in practice	A. Sweeney, D.W. Bustard	Software Quality Journal, Volume 6, Issue 4, Dec 1997, Pages 265 - 274
242	Software process modeling and enactment: an experience report related to problem tracking in an industrial project	Volker Gruhn, Juri Urbainczyk	Proceedings of the 20th international conference on Software engineering, Kyoto, Japan, Pages: 13 - 21, 1998
243	Software Process Models and Project Performance	M.S. Krishnan, Tridas Mukhopadhyay, Dave Zubrow	Information Systems Frontiers, Volume 1, Issue 3, Oct 1999, Pages 267 - 277
244	Software process simulation modeling: Why? What? How?	Marc I. Kellner, Raymond J. Madachy and David M. Raffo	Journal of Systems and Software, Volume 46, Issues 2-3, 15 April 1999, Pages 91-105
245	Software processes and project performance	Deephouse, Christopher, Mukhopadhyay, Tridas, Goldenson, Dennis R, Kellner, Marc I	Journal of Management Information Systems. Armonk: Winter 1995/1996. Vol. 12, Num. 3; pág. 187 (19 páginas)
246	Software quality and the Capability Maturity Model	James Herbsleb, David Zubrow, Dennis Goldenson, Will Hayes, Mark Paulk	Communications of the ACM, Volume 40 , Issue 6 (June 1997), Pages: 30 - 40, 1997
247	Software quality management and ISO 9000 implementation	Y Helio Yang	Industrial Management & Data Systems. Wembley: 2001. Vol. 101, Num. 7; pág. 329 (10 páginas)
248	Software Requirements Negotiation Using the Software Quality Function Deployment	João Ramires, Pedro Antunes, Ana Respício	Lecture Notes in Computer Science, Volume 3706, Oct 2005, Pages 308 - 324
249	Solid planning critical for client/server applications	Ballou, Melinda-Carol	Computerworld. Framingham: Oct 3, 1994. Vol. 28, Num. 40; pág. 83 (1 página)

250	SPI patterns: learning from experience	Blanco, M.; Gutierrez, P.; Satriani, G.	Software, IEEE , vol.18, no.3pp.28-35, May 2001
251	SPiCE in action - experiences in tailoring and extension	Cass, A.; Volcker, C.; Sutter, P.; Dorling, A.; Stienen, H.	Euromicro Conference, 2002. Proceedings. 28th , vol., no.pp. 352- 360, 2002
252	Standardization, requirements uncertainty and software project performance	Antonio Cicu	Information & Management, Volume 31, Issue 3, December 1996, Pages 135-150
253	Strategies for software reuse: a principal component analysis of reuse practices	Rothenberger, M.A.; Dooley, K.J.; Kulkarni, U.R.; Nada, N.	Software Engineering, IEEE Transactions on , vol.29, no.9pp. 825- 837, Sept. 2003
254	Structural change and change advocacy: a study in becoming a software engineering organization	Nelson, K.M.; Buche, M.; Nelson, H.J.,	System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on , vol., no.pp. 9 pp.-, 3-6 Jan. 2001
255	Successful process implementation	Borjesson, A.; Mathiassen, L.	Software, IEEE , vol.21, no.4pp. 36- 44, July-Aug. 2004
256	Supporting Adaptable Methodologies to Meet Evolving Project Needs	Scott Henninger, Aditya Ivaturi, Krishna Nuli, Ashok Thirunavukkaras	Lecture Notes in Computer Science, Volume 2418, Jan 2002, Pages 33 - 44
257	Table of contents	IEEE	Quality Software, 2003. Proceedings. Third International Conference on , vol., no.pp. 0_5- 0_8, 6- 7 Nov. 2003
258	Tailoring Cleanroom for industrial use	Oshana, R.S.	Software, IEEE , vol.15, no.6pp.46-55, Nov/Dec 1998
259	Teams in packaged software development: The Software Corp. experience	Line Dube	Information Technology & People. West Linn: 1998. Vol. 11, Num. 1; pág. 36
260	Technical communications as knowledge management: evolution of a profession	Patti Anklam	Proceedings of the 17th annual international conference on Computer documentation, New Orleans, Louisiana, United States, Pages: 36 - 44, 1999
261	Technology management and international cooperation: Several success stories	F G Filip, Adriana Alexandru, Irina Socol	Human Systems Management. Amsterdam: 1997. Vol. 16, Num. 3; pág. 223 (7 páginas)
262	Technology transfer macro-process: a practical guide for the effective introduction of technology	Tetsuto Nishiyama, Kunihiro Ikeda, Toru Niwa	Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, Pages: 577 - 586, 2000
263	Test-driven development concepts, taxonomy, and future direction	Janzen, D.; Saiedian, H.	Computer , vol.38, no.9pp. 43- 50, Sept. 2005
264	The AIAA/IEEE/SAE Digital Avionics Systems Conference	IEEE	Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE , vol.2, no.pp.i-xv, 31 Oct-7 Nov 1998
265	The appropriateness of Gutman's means-end chain model in software evaluation	Wong, B.	Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n , vol., no.pp. 56- 65, 2002
266	The Art and Science of Software Release Planning	Ruhe, G.; Saliu, M.O.	Software, IEEE , vol.22, no.6pp. 47- 53, Nov.-Dec. 2005
267	The benchmarking process: one team's experience	Fogle, S.; Loulis, C.; Neuendorf, B.,	Software, IEEE , vol.18, no.5pp.40-47, Sep/Oct 2001
268	The case for re-examining IT effectiveness	Priya Kurien, Was Rahman, V S Purushottam	The Journal of Business Strategy. Boston: 2004.Vol.25, Num. 2; pg. 29, 8 pgs
269	The Collaboration Engineering Maturity Model	Santanen, E.; Kolfshoten, G.; Golla, K.	System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on , vol.1, no.pp. 16c- 16c, 04-07 Jan. 2006
270	The evolution of goal-based information modelling: literature review	Andrew John Boyd	Aslib Proceedings. Bradford: 2005.Vol.57, Num. 6; pg. 523, 16 pgs
271	The evolution of quality processes at Tata Consultancy Services	Kenni, G.	Software, IEEE , vol.17, no.4pp.79-88, Jul/Aug 2000
272	The Evolving Object of Software Development	Paul S. Adler	Organization, May 2005; 12: 401 - 435.
273	The impact of Ada and object-oriented design in NASA Goddard's Flight Dynamics Division	Sharon Waligora, John Bailey, Mike Stark	ACM SIGAda Ada Letters, Volume XVII Issue 3, Pages: 67 - 86, 1997
274	The impact of critical success factors across the stages of enterprise resource planning implementations	Somers, T.M.; Nelson, K.	System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on , vol., no.pp. 10 pp.-, 3-6 Jan. 2001
275	The impact of goals on software project management: An experimental investigation	Tarek K Abdel-Hamid, Kishore Sengupta, Clint Swett	MIS Quarterly. Minneapolis: Dec 1999. Vol. 23, Num. 4; pág. 531 (25 páginas)
276	The impacts of quality and productivity perceptions on the use of software process improvement innovations	Gina C. Green, Alan R. Hevner and Rosann Webb Collins	Information and Software Technology, Volume 47, Issue 8, 1 June 2005, Pages 543-553
277	The Individual Deployment of Systems Development Methodologies	Magda Huisman, Juhani Iivari	Lecture Notes in Computer Science, Volume 2348, Jan 2002, Page 134
278	The Internal Consistencies of the 1987 SEI Maturity Questionnaire and the SPICE Capability Dimension	Pierfrancesco Fusaro, Khaled El Emam, Bob Smith	Empirical Software Engineering, Volume 3, Issue 2, Jun 1998, Pages 179 - 201
279	The internal consistency of the ISO/IEC 15504 software process capability scale	El Emam, K.	Software Metrics Symposium, 1998. Metrics 1998. Proceedings. Fifth International , vol., no.pp.72-81, 20-21 Nov 1998
280	The Personal Software Process (PSP) tutorial	Humphrey, W.; Over, J.	Software Engineering Workshop, 2002. Proceedings. 27th Annual NASA Goddard , vol., no.pp. 39- 92, 2002

281	The process workshop: a tool to define electronic process guides in small software companies	Dingsoyr, T.; Moe, N.B.	Software Engineering Conference, 2004. Proceedings. 2004 Australian , vol., no.pp. 350- 357, 2004
282	The simulation of formal management and technical reviews to increase quality in undergraduate software engineering projects	Terry, J.E.; Hope, S.	Software Engineering: Education and Practice, 1998. Proceedings. 1998 International Conference , vol., no.pp.113-119, 26-29 Jan 1998
283	The successful diffusion of innovations: guidance for software development organizations	Green, G.C.; Hevner, A.R.,	Software, IEEE , vol.17, no.6pp.96-103, Nov/Dec 2000
284	The Utilization of Information Systems Development Methodologies in Practice	Karlheinz Kautz, Bo Hansen, Dan Jacobsen	Journal of Information Technology Cases and Applications. Marietta: 2004.Vol.6, Num. 4; pg. 1, 20 pgs
285	TIM: a tool insertion method	Tilmann Bruckhaus	Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada, Page: 7, 1994
286	Tool Support for Experience-Based Methodologies	Scott Henninger	Lecture Notes in Computer Science, Volume 2640, Jan 2003, Pages 44 - 59
287	Total quality management in information systems development: Key constructs and relationships	T Ravichandran, Arun Rai	Journal of Management Information Systems. Armonk: Winter 1999/2000. Vol. 16, Num. 3; pág. 119 (37 páginas)
288	Total software process model evolution in EPOS: experience report	Minh N. Nguyen, Alf Inge Wang, Reidar Conradi	Proceedings of the 19th international conference on Software engineering, Boston, Massachusetts, United States, Pages: 390 - 399, 1997
289	Toward an architectural knowledge base for wireless service engineering	Niemela, E.; Kalaaja, J.; Lago, P.	Software Engineering, IEEE Transactions on , vol.31, no.5pp. 361- 379, May 2005
290	Towards maturity of information security maturity criteria: Six lessons learned from software maturity criteria	Mikko Siponen	Information Management & Computer Security. Bradford: 2002. Vol. 10, Num. 5; pág. 210 (15 páginas)
291	Understanding and evaluating collaborative work in multi-site software projects - a framework proposal and preliminary results	Martiin, P.; Lehto, J.A.; Nyman, G.	System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on , vol., no.pp. 283- 292, 7-10 Jan. 2002
292	Understanding software defect detection in the Personal Software Process	Wohlin, C.; Wesslen, A.	Software Reliability Engineering, 1998. Proceedings. The Ninth International Symposium on , vol., no.pp.49-58, 4-7 Nov 1998
293	Understanding the implementation of software process improvement innovations in software organizations	Karlheinz Kautz, Peter Axel Nielsen	Information Systems Journal, Volume 14, Issue 1, Page 3-22, Jan 2004
294	University/industry collaboration in developing a simulation-based software project management training course	Collofello, J.S.	Education, IEEE Transactions on , vol.43, no.4pp. 389- 393, Nov 2000
295	Usability in Practice: user experience lifecycle - evolution and revolution	Stephanie Rosenbaum, Chauncey E. Wilson, Timo Jokela, Janice A. Rohn, Trixi B. Smith, Karel Vredenburg	CHI '02 extended abstracts on Human factors in computing systems, Minneapolis, Minnesota, USA, Pages: 898 - 903, 2002
296	Using ABC Model for Software Process Improvement: A Balanced Perspective	Han-Wen Tuan; Chia-Yi Liu; Chiou-Mei Chen	System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on , vol.9, no.pp. 229c- 229c, 04-07 Jan. 2006
297	Using an expert panel to validate a requirements process improvement model	Sarah Beecham, Tracy Hall, Carol Britton, Michaela Cottee and Austen Rainer	Journal of Systems and Software, Volume 76, Issue 3, June 2005, Pages 251-275
298	Using empirical knowledge from replicated experiments for software process simulation: a practical example	Munch, J.; Armbrust, O.	Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on , vol., no.pp. 18- 27, 30 Sept.-1 Oct. 2003
299	Using Empirical Studies during Software Courses	Jeffrey Carver, Letizia Jaccheri, Sandro Morasca, Forrest Shull	Lecture Notes in Computer Science, Volume 2765, Aug 2003, Page 81
300	Using group support systems for software inspections	van Genuchten, M.; van Dijk, C.; Scholten, H.; Vogel, D.	Software, IEEE , vol.18, no.3pp.60-65, May 2001
301	Using simulation to analyse the impact of software requirement volatility on project performance	D. Pfahl and K. Lebsanft	Information and Software Technology, Volume 42, Issue 14, 15 November 2000, Pages 1001-1008
302	Using simulation to build inspection efficiency benchmarks for development projects	Lionel Briand, Khaled El Emam, Oliver Laitenberger, Thomas Fussbroich	Proceedings of the 20th international conference on Software engineering, Kyoto, Japan, Pages: 340 - 349, 1998
303	Using Simulation to Visualise and Analyse Product-Process Dependencies in Software Development Projects	Dietmar Pfahl, Andreas Birk	Lecture Notes in Computer Science, Volume 1840, Jan 2000, Page 88
304	Using the focus group method in software engineering: obtaining practitioner and user experiences	Kontio, J.; Lehtola, L.; Bragge, J.	Empirical Software Engineering, 2004. ISESE '04. Proceedings. 2004 International Symposium on , vol., no.pp. 271- 280, 19-20 Aug. 2004
305	Using the software capability maturity model for certification projects	Rierson, L.K.	Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE , vol.1, no.pp.C24/1-C24/8 vol.1, 31 Oct-7 Nov 1998
306	Validating the ISO/IEC 15504 measure of software requirements analysis process capability	El Emam, K.; Birk, A.	Software Engineering, IEEE Transactions on , vol.26, no.6pp.541-566, Jun 2000
307	Value-based software engineering: reinventing	Barry Boehm	ACM SIGSOFT Software Engineering Notes, Volume 28 , Issue 2 (March 2003), Page: 3, 2003

308	VDM over PSP: a pilot course for VDM beginners to confirm its suitability for their development	Suzumori, H.; Kaiya, H.; Kajiri, K.	Computer Software and Applications Conference, 2003. COMPSAC 2003. Proceedings. 27th Annual International , vol., no.pp. 327- 334, 3-6 Nov. 2003
309	Verification and validation in industry - a qualitative survey on the state of practice	Andersson, C.; Runeson, P.	Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n , vol., no.pp. 37- 47, 2002
310	Victor R. Basili's Contributions to Software Quality	Shull, F.; Seaman, C.; Zelkowitz, M.	Software, IEEE , vol.23, no.1pp. 16- 18, Jan.-Feb. 2006
311	Virtual Software Engineering Laboratories in Support of Trade-off Analyses	Jürgen Münch, Dietmar Pfahl, Ioana Rus	Software Quality Journal, Volume 13, Issue 4, Dec 2005, Pages 407 - 428
312	What do software practitioners really think about project success: an exploratory study	J. Drew Procaccino, June M. Verner, Katherine M. Shelfer and David Gefen	Journal of Systems and Software, Volume 78, Issue 2, November 2005, Pages 194-203
313	What makes measuring software so hard?	Rifkin, S.	Software, IEEE , vol.18, no.3pp.41-45, May 2001
314	When Management Gets Serious About Managing Software	Jansma, P.A.T.	Aerospace, 2005 IEEE Conference , vol., no.pp. 1- 17, 05-12 March 2005
315	Work outcomes and job design for contract versus permanent information systems professionals on software development teams	Soon Ang, Sandra A Slaughter	MIS Quarterly. Minneapolis: Sep 2001. Vol. 25, Num. 3; pág. 321 (30 páginas)
316	Workflow management based on process model repositories	Volker Gruhn, Monika Schneider	Proceedings of the 20th international conference on Software engineering, Kyoto, Japan, Pages: 379 - 388, 1998

## APÊNDICE E - Publicações com restrições de acesso nas bibliotecas digitais

Título	Autoria	Fonte
(Continuing) education of software professionals	Ilkka J. Haikala, Jukka Marjarvi	Lecture Notes in Computer Science, Volume 640, Jul 1992, Pages 180 - 193
A Framework for Classification of Change Approaches Based on a Comparison of Process Improvement Models	Otto Vinter	Lecture Notes in Computer Science, Volume 3547, Jun 2005, Pages 29 - 38
A Meta-model for Requirements Engineering in System Family Context for Software Process Improvement Using CMMI	Rodrigo Cerón, Juan C. Dueñas, Enrique Serrano, Rafael Capilla	Lecture Notes in Computer Science, Volume 3547, Jun 2005, Pages 173 - 188
A Model for the Implementation of Software Process Improvement: An Empirical Study	Mahmood Niazi, David Wilson, Didar Zowghi, Bernard Wong	Lecture Notes in Computer Science, Volume 3009, Jan 2004, Pages 1 - 16
A Practical Application of the IDEAL Model	Valentine Casey, Ita Richardson	Lecture Notes in Computer Science, Volume 2559, Jan 2002, Pages 172 - 184
A Unified Approach for Software Policy Modeling: Incorporating Implementation into a Modeling Methodology	Junho Shim, Seungjin Lee, Chisu Wu	Lecture Notes in Computer Science, Volume 2813, Jan 2003, Pages 118 - 130
Active Probes Synergy in Experience-Based Process Improvement	Kurt Schneider	Lecture Notes in Computer Science, Volume 1840, Jan 2000, Page 6
An Experimental Replica to Validate a Set of Metrics for Software Process Models	Félix García, Francisco Ruiz, Mario Piattini	Lecture Notes in Computer Science, Volume 3281, Jan 2004, Pages 79 - 90
An Overview of Process Technology Activities at the Software Engineering Institute	Alan M. Christie, Marc I. Kellner, William E. Riddle	Journal of Systems Integration, Volume 8, Issue 2, May 1998, Pages 121 - 132
Are the Perspectives Really Different? – Further Experimentation on Scenario-Based Reading of Requirements	Björn Regnell, Per Runeson, Thomas Thelin	Empirical Software Engineering, Volume 5, Issue 4, Dec 2000, Pages 331 - 356
Brazilian Software Process Reference Model and Assessment Method	Kival C. Weber, Eratóstenes E.R. Araújo, Ana Regina C. Rocha, Cristina A.F. Machado, Danilo Scalet, Clénio F. Salviano	Lecture Notes in Computer Science, Volume 3733, Nov 2005, Pages 402 - 411
Canada: meeting the software quality challenge; current activities	P. Voldner	Software Quality Journal (Historical Archive), Volume 2, Issue 4, Dec 1993, Pages 239 - 243
Case study in innovative process improvement: Code synthesis from formal specifications	Garbett, P.; Parkes, J.P.; Shackleton, M.; Anderson, S.	Microprocessors and Microsystems, v 23, n 7, Dec, 1999, p 417-424
Cognitive Structures of Software Evaluation: A Means-End Chain Analysis of Quality	Bernard Wong, Ross Jeffery	Lecture Notes in Computer Science, Volume 2188, Jan 2001, Page 6
Creating a software process improvement program	B Curtis and M Paulk	Information and Software Technology, Volume 35, Issues 6-7, June-July 1993, Pages 381-386
Daibutsu-den: A Component-Based Framework for Organizational Process Asset Utilization	Hajimu Iida, Yasushi Tanaka, Ken'ichi Matsumoto	Lecture Notes in Computer Science, Volume 2559, Jan 2002, Pages 207 - 219
Enabling Local SPI in a Multi-national Company	Peter Fröhlich, Horst Lichter, Manfred Zeller	Lecture Notes in Computer Science, Volume 2188, Jan 2001, Page 227
Enterprise-Oriented Software Development Environments to Support Software Products and Processes Quality Improvement	Mariano Montoni, Gleison Santos, Karina Villela, Ana Regina Rocha, Guilherme H. Travassos, Sávio Figueiredo, Sômulo Mafra, Adriano Albuquerque, Paula Mian	Lecture Notes in Computer Science, Volume 3547, Jun 2005, Pages 370 - 384
e-R&D – Effectively Managing Process Diversity	Christof Ebert, Jozef De Man	Annals of Software Engineering, Volume 14, Issue 1 - 4, Dec 2002, Pages 73 - 91
ERW'97 Session Report: Reuse Adoption Experiences Across a Large Corporation	Sergio Bandinelli	Lecture Notes in Computer Science, Volume 1429, Jan 1998, Page 249
Evaluating inheritance depth on the maintainability of object-oriented software	John Daly, Andrew Brooks, James Miller, Marc Roper, Murray Wood	Empirical Software Engineering (Historical Archive), Volume 1, Issue 2, Jan 1996, Pages 109 - 132
Experience Based Process Improvement	Kurt Schneider	Lecture Notes in Computer Science, Volume 2349, Jan 2002, Page 114, Publisher: Springer-Verlag London, UK
Experiences in transferring a software process improvement methodology to production laboratories	Michael Halliday, Inderpal Bhandari, Jarir Chaar and Ram Chillarege	Journal of Systems and Software, Volume 26, Issue 1, July 1994, Pages 61-68
Experiences with the capability maturity model in a research environment	M. J. Velden, J. Vreke, B. Wal, A. Symons	Software Quality Journal, Volume 5, Issue 2, Jun 1996, Pages 87 - 95
GARP - The Evolution of a Software Acquisition Process Model	Thomas Gantner, Tobias Häberlein	Lecture Notes in Computer Science, Volume 2349, Jan 2002, Page 186
Improving IT effectiveness through software process assessment	Mac Craigmyle, Irwin Fletcher	Software Quality Journal (Historical Archive), Volume 2, Issue 4, Dec 1993, Pages 257 - 264
Improving Knowledge Management in Software Reuse Process	Timo Kucza, Minna Nättinen, Päivi Parviainen	Lecture Notes in Computer Science, Volume 2188, Jan 2001, Page 141
Improving reliability of large software systems	Christof Ebert, Thomas Liedtke, Ekkehard Baisch	Annals of Software Engineering, Volume 8, Issue 1 - 4, Mar 1999, Pages 3 - 51
Improving the Express Process Appraisal Method	F. McCaffery, D. McFall, F.G. Wilkie	Lecture Notes in Computer Science, Volume 3547, Jun 2005, Pages 29 - 38

Improving Validation Activities in a Global Software Development	Christof Ebert, Casimiro Hernandez Parro, Roland Suttels, Harald Kolarczyk	Lecture Notes in Computer Science, Volume 2006, Jan 2001, Page 83
Industrial Experiences from SCM Current State Analysis	Tua Rahikkala, Jorma Taramaa, Antti Välimäki	Lecture Notes in Computer Science, Volume 1439, Jan 1998, Page 13
Is this training? A unique approach to software process training in industry	Pauline D. Fortin, Christine A. Jeske, John R. Lakey, Kristin B. Urquhart, Anthony Vea	Lecture Notes in Computer Science, Volume 895, Jun 1995, Pages 409 - 417
Knowledge Mapping: A Technique for Identifying Knowledge Flows in Software Organisations	Bo Hansen Hansen, Karlheinz Kautz	Lecture Notes in Computer Science, Volume 3281, Jan 2004, Pages 126 - 137
Management of Workflow over the Web Supporting Distributed Process Evolution	Sang-Yoon Min, Doo-Hwan Bae, Sung-Chan Cho, Young-Kwang Nam	Lecture Notes in Computer Science, Volume 1749, Jan 1999, Pages 367 - 372
Managing the Software Process in the Middle of Rapid Growth: A Metrics Based Experiment Report from Nokia	Tapani Kilpi	Lecture Notes in Computer Science, Volume 1789, Jan 2000, Page 498
Maturing XP through the CMM	Jonas Martinsson	Lecture Notes in Computer Science, Volume 2675, Jan 2003, Pages 80 - 87
Models and Success Factors of Process Change	Marion Lepasaar, Timo Varkoi, Hannu Jaakkola	Lecture Notes in Computer Science, Volume 2188, Jan 2001, Page 68
Organizational Transition to Object Technology: Theory and Practice	M.K. Serour, B. Henderson-Sellers, J. Hughes, D. Winder, L. Chow	Lecture Notes in Computer Science, Volume 2425, Jan 2002, Page 229
Petri Nets and Software Engineering	Giovanni Denaro, Mauro Pezzè	Lecture Notes in Computer Science, Volume 3098, Jan 2004, Pages 439 - 466
Process Failure in a Rapidly Changing High-Tech Organisation: A System Dynamics View	Bruce Campbell	Lecture Notes in Computer Science, Volume 1507, Jan 1998, Pages 291 - 300
Process Metamodelling and Process Construction: Examples Using the OPEN Process Framework (OPF)	Brian Henderson-Sellers	Annals of Software Engineering, Volume 14, Issue 1 - 4, Dec 2002, Pages 341 - 362
Reference Model for Software Process Improvement: A Brazilian Experience	Ana Regina Rocha, Mariano Montoni, Gleison Santos, Sômulo Mafra, Sávio Figueiredo, Adriano Albuquerque, Paula Mian	Lecture Notes in Computer Science, Volume 3792, Oct 2005, Pages 130 - 141
Reference Model for Software Process Improvement: A Brazilian Experience	Rocha AR, Montoni M, Santos G, et al.	Lecture Notes in Computer Science 3792: 130-141 2005
Requirements traceability: Theory and practice	Balasubramaniam Ramesh, Curtis Stubbs, Timothy Powers, Michael Edwards	Annals of Software Engineering, Volume 3, Issue 0, Jan 1997, Pages 397 - 415
Reuse Based Software Factory	Manu Prego	Lecture Notes in Computer Science, Volume 2765, Aug 2003, Pages 256 - 273
"Software Development Bug Tracking: ""Tool Isn't User Friendly"" or ""User Isn't Process Friendly""	Leah Goldin, Lilach Rochell	Lecture Notes in Computer Science, Volume 2349, Jan 2002, Page 226
Software Process Improvement Motivators: An Analysis using Multidimensional Scaling	Nathan Baddoo, Tracy Hall	Empirical Software Engineering, Volume 7, Issue 2, Jun 2002, Page 93
Software Process Improvement through Use Cases: Building Quality from the Very Beginning	Andrea Valerio, Massimo Fenaroli	Lecture Notes in Computer Science, Volume 2559, Jan 2002, Pages 398 - 406
Software process maturity (SPM) and the information systems developer	K Thompson and P McParland	Information and Software Technology, Volume 35, Issues 6-7, June-July 1993, Pages 331-338
Software Product Family Evaluation	Frank van der Linden, Jan Bosch, Erik Kamsties, Kari Känvälä, Lech Krzanik, Henk Obbink	Lecture Notes in Computer Science, Volume 3014, Jan 2004, Pages 352 - 369
SPICE for space - A method of process assessment for space software projects	Dorling, A.; Voelcker, C.; Cass, A.; Winzer, L.	DASIA 2001 - Data Systems in Aerospace; Proceedings of the Conference. 2001
Studying Software Engineers: Data Collection Techniques for Software Field Studies	Timothy C. Lethbridge, Susan Elliott Sim, Janice Singer	Empirical Software Engineering, Volume 10, Issue 3, Jul 2005, Pages 311 - 341
The Future of Software Processes	Barry Boehm	Lecture Notes in Computer Science, Volume 3840, Dec 2006, Pages 10 - 24
The Human Dimension of the Software Process	David Wastell, Selma Arbaoui, Jacques Lonchamp, Carlo Montangero	Lecture Notes in Computer Science, Volume 1500, Jan 1999, Page 165
The introduction of an Object Oriented Analysis/Design method and Object Oriented Metrics in the software development life-cycle	Rik Simoens	Lecture Notes in Computer Science, Volume 1031, Jun 1996, Pages 271 - 283
The PROFES Improvement Methodology - Enabling Technologies and Methodology Design	Janne Järvinen, Seija Komi-Sirviö, Guenther Ruhe	Lecture Notes in Computer Science, Volume 1840, Jan 2000, Pages 257 - 270
The value of a certified quality management system: the perception of internal developers	K. Robinson, P. Simmons	Software Quality Journal, Volume 5, Issue 2, Jun 1996, Pages 61 - 73
Towards Comprehensive Experience-Based Decision Support	Andreas Jedlitschka, Dietmar Pfahl	Lecture Notes in Computer Science, Volume 3281, Jan 2004, Pages 34 - 45
Transition Management of Software Process Improvement	Seon-ah Lee, Byoungju Choi	Lecture Notes in Computer Science, Volume 2559, Jan 2002, Pages 19 - 34
Transitioning to the CMMI -SE/SW - An evolutionary process	Menezes, Winifred	Systems Engineering, v 5, n 1, February, 2002, p 32-40
Using Evolutionary Project Management (Evo) to Create Faster, More Userfriendly and More Productive Software. Experience Report from FIRM AS, a Norwegian Software Company	Trond Johansen	Lecture Notes in Computer Science, Volume 3547, Jun 2005, Pages 216 - 223
Using Measurement Data in a TSP <sup>SM</sup> Project	Noopur Davis, Julia Mullaney, David Carrington	Lecture Notes in Computer Science, Volume 3281, Jan 2004, Page 91

## APÊNDICE F - Publicações selecionadas no 1º filtro

Título	Autoria	Dados da Publicação
A case study of software process improvement during development	Bhandari, I.; Halliday, M.; Tarver, E.; Brown, D.; Chaar, J.; Chillarege, R.	Software Engineering, IEEE Transactions on , vol.19, no.12pp.1157-1170, Dec 1993
A procedure for customizing a software process	Ibarguengoitia, G.; Salazar, J.A.; Sanchez, M.G.; Ramirez, A.Y.	Computer Science, 2003. ENC 2003. Proceedings of the Fourth Mexican International Conference on , vol., no.pp. 68- 72, 8-12 Sept. 2003
A systematic survey of CMM experience and results	James D. Herbsleb, Dennis R. Goldenson	Proceedings of the 18th international conference on Software engineering, Berlin, Germany, Pages: 323 - 330, 1996
An Industrial Case Study of Implementing and Validating Defect Classification for Process Improvement and Quality Management	Freimut, B.; Denger, C.; Ketterer, M.	Software Metrics, 2005. 11th IEEE International Symposium , vol., no.pp. 19- 19, 19-22 Sept. 2005
Designing and Conducting an Empirical Study on Test Management Automation	Griselda Girauo, Paolo Tonella	Empirical Software Engineering, Volume 8, Issue 1, Mar 2003, Pages 59 - 81
Empirically Driven Design of Software Development Processes for Wireless Internet Services	Ulrike Becker-Kornstaedt, Daniela Boggio, Jürgen Münch, Alexis Ocampo, Gino Palladino	Lecture Notes in Computer Science, Volume 2559, Jan 2002, Pages 351 - 366
European experiences with software process improvement	Fran O'Hara	Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, Pages: 635 - 640, 2000
Evaluation of code review methods through interviews and experimentation	Martin Höst and Conny Johansson	Journal of Systems and Software, Volume 52, Issues 2-3, 1 June 2000, Pages 113-120
Evidence-based software engineering for practitioners	Dyba, T.; Kitchenham, B.A.; Jorgensen, M.	Software, IEEE , vol.22, no.1pp. 58- 65, Jan.- Feb. 2005
Experience with software process in physics projects	Guatelli, S.; Mascialino, B.; Moneta, L.; Papadopoulos, I.; Pfeiffer, A.; Pia, M.G.; Piergentili, M.	Nuclear Science Symposium Conference Record, 2004 IEEE , vol.4, no.pp. 2100- 2103 Vol. 4, 16-22 Oct. 2004
Experiences from the pilot operation and commissioning phase of a SCM process improvement program	Nattinen, M.; Rahikkala, T.; Valimaki, A.,	EUROMICRO Conference, 1999. Proceedings. 25th , vol.2, no.pp.185-192 vol.2, 1999
Experiences in the Application of Software Process Improvement in SMES	Jose A. Calvo-Manzano Villalón, Gonzalo Cuevas Agustín, Tomás San Feliu Gilabert, Antonio De Amescua Seco, et al.	Software Quality Journal. London: Nov 2002. Vol. 10, Num. 3; pág. 261
Experiences in the Application of Software Process Improvement in SMES	Jose A. Calvo-Manzano Villalón, Gonzalo Cuevas Agustín, Tomás San Feliu Gilabert, Antonio De Amescua Seco, Luis García Sánchez, Manuel Pérez Cota	Lecture Notes in Computer Science, Volume 3547, Jun 2005, Pages 216 - 223
IEEE Std 1348-1995. IEEE recommended practice for the adoption of Computer-Aided Software Engineering (CASE) tools	IEEE	IEEE Std 1348-1995 , vol., no.pp.-, 10 Apr 1996
Implementing requirements engineering processes throughout organizations: success factors and challenges	Marjo Kauppinen, Matti Vartiainen, Jyrki Kontio, Sari Kujala and Reijo Sulonen	Information and Software Technology, Volume 46, Issue 14, 1 November 2004, Pages 937-953
Improving diffusion practices in a software organization	Andersson, I.; Nilsson, K.	System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on , vol., no.pp. 8 pp.-, 7-10 Jan. 2002
Improving Validation Activities in a Global Software Development	Christof Ebert, Casimiro Hernandez Parro, Roland Suttels, Harald Kolarczyk	Proceedings of the 23rd International Conference on Software Engineering, Toronto, Ontario, Canada, Pages: 545 - 554, 2001
Introducing requirements engineering: how to make a cultural change happen in practice	Kauppinen, M.; Kujala, S.; Aaltio, T.; Lehtola, L.	Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on , vol., no.pp. 43- 51, 2002
Knowledge Support in Software Process Tailoring	Peng Xu	System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on , vol., no.pp. 87c- 87c, 03-06 Jan. 2005
Lessons Learned and Recommendations from Two Large Norwegian SPI Programmes	Reidar Conradi Tore Dybå1, Dag I.K. Sjøberg and Tor Ulsund	Lecture Notes in Computer Science, Volume 2786, Jan 2003, Pages 32 - 45
Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory	Victor R. Basili, Frank E. McGarry, Rose Pajerski, Marvin V. Zelkowitz	Proceedings of the 24th International Conference on Software Engineering, Orlando, Florida, Pages: 69 - 79, 2002
Lessons learned implementing ISO 9001 in a software organization	Ganner, M.; Johnson, M.	Innovation in Technology Management - The Key to Global Leadership. PICMET '97: Portland International Conference on Management and Technology , vol., no.pp.606-608, 27-31 Jul 1997
Lessons learned in framework-based software process improvement	Jalote, P.	Software Engineering Conference, 2002. Ninth Asia-Pacific , vol., no.pp. 261- 265, 2002

Título	Autoria	Dados da Publicação
Measurement processes are software, too	M. Morisio	Journal of Systems and Software, Volume 49, Issue 1, 15 December 1999, Pages 17-31
Measurement program success factors revisited	Frank Niessink and Hans van Vliet	Information and Software Technology, Volume 43, Issue 10, 15 August 2001, Pages 617-628
Process improvement and the corporate balance sheet	Dion, R.	Software, IEEE , vol.10, no.4pp.28-35, Jul 1993
Process improvement for small organizations	Kelly, D.P.; Culleton, B.	Computer , vol.32, no.10pp.41-47, Oct 1999
Siemens process assessment and improvement approaches: experiences and benefits	Mehner, T.; Messer, T.; Paul, P.; Paulisch, F.; Schless, P.; Volker, A.	Computer Software and Applications Conference, 1998. COMPSAC '98. Proceedings. The Twenty-Second Annual International , vol., no.pp.186-195, 19-21 Aug 1998
Software Development and Experimentation in an Academic Environment: The Gaudi Experience	Ralph-Johan Back, Luka Milovanov, Ivan Porres	Lecture Notes in Computer Science, Volume 3547, Jun 2005, Pages 414 - 428
Software process improvement as organizational knowledge creation: a multiple case analysis	Arent, J.; Norbjerg, J.	System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on , vol., no.pp. 11 pp.-, 4-7 Jan. 2000
Software process improvement experience in the DP/MIS function: experience report	Albert Johnson	Proceedings of the 16th international conference on Software engineering, Sorrento, Italy, Pages: 323 - 329, 1994
Teams in packaged software development: The Software Corp. experience	Line Dubé	Information Technology & People; Volume: 11 Issue: 1; 1998
Technology transfer macro-process: a practical guide for the effective introduction of technology	Tetsuto Nishiyama, Kunihiro Ikeda, Toru Niwa	Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, Pages: 577 - 586, 2000
The impact of Ada and object-oriented design in NASA Goddard's Flight Dynamics Division	Sharon Waligora, John Bailey, Mike Stark	ACM SIGAda Ada Letters, Volume XVII Issue 3, Pages: 67 - 86, 1997
TIM: a tool insertion method	Tilman Bruckhaus	Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada, Page: 7, 1994
Value-based software engineering: reinventing	Barry Boehm	ACM SIGSOFT Software Engineering Notes, Volume 28 , Issue 2 (March 2003), Page: 3, 2003

# APÊNDICE G - Modelo de processo para realização de projetos-piloto

---

Esta seção descreve de forma detalhada o processo definido para apoiar a realização (planejamento, execução e análise) de projetos-piloto.

## Objetivo

Avaliar a efetividade de propostas de melhoria de processo de software através da realização de projetos-piloto.

## Insumo do Processo

Proposta de melhoria de processos de software.

## Critérios de Entrada

Proposta de melhoria contendo os objetivos de melhoria, os resultados esperados com a aplicação das ações de melhoria no(s) processo(s), a descrição detalhada das ações de melhoria e o contexto ao qual as melhorias se aplicam (tipo, tamanho e duração de projeto, paradigma e outras que forem necessárias).

## Produto do Processo

Laudo de avaliação da proposta de melhoria de processos de software.

## Critério de Saída

O laudo deve conter a identificação dos projetos-piloto envolvidos na avaliação, síntese quantitativa e parecer final devidamente justificado informando o resultado da avaliação da proposta de melhoria.

## Papéis envolvidos

Grupo de Processo: Pessoas responsáveis pela definição e melhoria dos processos e ativos de processo da organização.

Gerente do Projeto: Responsável pela condução das atividades dos projetos desenvolvidos pela organização.

Equipe do Projeto: Pessoas que participam do projeto. Dentre eles: analistas, testadores, programadores, arquitetos de software, projetistas, dentre outros.

## Atividades

### 1. Definir Piloto

Esta primeira macroatividade consiste na identificação do piloto, definição dos objetivos, descrição das melhorias a serem avaliadas, definição de um plano de medição e declaração das expectativas para o projeto piloto.

#### 1.1. Identificar Piloto

**Descrição:** Identificar o piloto através de um nome que represente a mudança a ser testada no processo, informar o tipo de processo (padrão, especializado ou instanciado) que será submetido a mudança, determinar quantos projetos serão utilizados para testar a mudança, identificar qual o processo que será alterado, declarar o objetivo do piloto e quais os resultados esperados para a avaliação (expectativa geral).

**Artefatos Requeridos:** Proposta de melhoria de processos de software.

**Critérios de Entrada:** Proposta de melhoria contendo objetivos de melhoria, os resultados esperados com a aplicação das ações de melhoria no processo e a informação sobre o processo a ser modificado.

**Artefatos Produzido:** Identificação do piloto

**Critérios de Saída:** Todos os atributos relacionados à identificação do piloto devem ter sido declarados.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

### **1.2. Descrever Melhoria para Avaliação**

**Descrição:** Realizar o registro das ações de melhoria a serem realizadas no processo. Estas melhorias podem consistir na alteração de atividades (inclusão, alteração, exclusão ou reordenação), utilização de uma nova ferramenta de apoio ao desenvolvimento, utilização de novos métodos e/ou técnicas na execução de determinadas atividades, alteração ou inserção de novos roteiros.

**Artefatos Requeridos:** Proposta de melhoria de processos de software

**Crítérios de Entrada:** A proposta de melhoria deve conter as mudanças que serão aplicadas no processo (ações de melhoria).

**Artefatos Produzidos:** Lista das ações de melhoria que serão testadas nos projetos.

**Crítérios de Saída:** Ao menos uma ação de melhoria deve ser descrita.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Identificar Piloto

### **1.3. Elaborar Plano de Medição para Avaliação da Melhoria**

**Descrição:** Identificar os objetivos, questões e métricas para relacionadas à avaliação das melhorias no projeto piloto. As métricas selecionadas servirão a dois propósitos: mensurar os efeitos obtidos com a aplicação das alterações no processo e permitir o monitoramento de fatores mensuráveis que podem influenciar os resultados da avaliação.

**Artefatos Requeridos:** Proposta de melhoria de processos de software

**Artefatos Produzidos:** Plano de medição para avaliação da proposta de melhoria

**Crítérios de Saída:** O plano deve possuir ao menos uma métrica.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Descrever Melhoria para Avaliação

### **1.4. Registrar Expectativas da Avaliação**

**Descrição:** Para cada métrica responsável pela mensuração dos efeitos das mudanças no processo, estabelecer um valor que represente as expectativas relativas à aplicação das melhorias. Este valor pode ser obtido a partir da linha base da métrica na organização, das expectativas dos participantes (*survey*), *benchmarking* ou opinião de especialistas. A declaração das expectativas deve ser justificada, informando inclusive a fonte dos valores esperados para cada métrica.

**Artefatos Requeridos:** Plano de medição para avaliação da proposta de melhoria.

**Crítérios de Entrada:** Possuir ao menos uma métrica identificada no plano de medição.

**Artefatos Produzidos:** Expectativas da avaliação

**Crítérios de Saída:** Ao menos uma expectativa (valor para métrica, origem do valor e justificativa) deve ser declarada.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Elaborar Plano de Medição para Avaliação da Melhoria

## **2. Definir Diretrizes para Realização do Piloto**

A execução desta macroatividade resultará nas diretrizes que apoiaram a seleção do projeto a ser utilizado como piloto para avaliar a melhoria. Também está inclusa a atividade que apoiará a identificação dos fatores que podem exercer alguma influência durante a execução do projeto de forma a dificultar a avaliação quantitativa e/ou qualitativa das melhorias a serem testadas.

### **2.1. Estabelecer Critérios para Seleção do Projeto**

**Descrição:** Estabelecer os critérios de seleção do projeto a ser utilizado como piloto e dos participantes da equipe do projeto. Há casos que a avaliação somente poderá ser realizada se o projeto apresentar uma determinada característica. Exemplo: se no processo de desenvolvimento de software foi inserida uma atividade que se aplica apenas ao desenvolvimento de softwares para web, o projeto-piloto deverá possuir tal característica. Um exemplo relativo aos critérios dos participantes: caso a organização queira avaliar o desempenho do grupo de trabalho com o uso de uma nova tecnologia,

é necessário selecionar participantes que tenham o mesmo nível de conhecimento para a tecnologia anterior.

**Artefatos Requeridos:** Proposta de melhoria de processos de software

**Critérios de Entrada:** A proposta de melhoria deve identificar as características dos projetos (tamanho, esforço, tipo de equipe, experiência da gerência e outras que forem relevantes) ao qual a mudança do processo se aplica. Caso a proposta seja aplicável a qualquer tipo de projeto, é necessário declarar esta informação explicitamente.

**Artefatos Produzidos:** Critérios para seleção do projeto

**Critérios de Saída:** Ao menos um critério deve ter sido declarado, do contrário, a informação que a proposta se aplica a qualquer projeto deve ser explicitada.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Registrar Expectativas da Avaliação

## 2.2. Identificar Fatores de Influência

**Descrição:** Identificar os fatores que podem influenciar direta ou indiretamente nos resultados esperados (expectativas) e identificar maneiras de isolar ou identificar seus efeitos, de forma a não influenciar na análise ou, ao menos, permitir a realização de uma análise qualitativa ciente da existência destes fatores.

**Artefatos Requeridos:** Proposta de melhoria

**Artefatos Produzidos:** Lista de fatores de influência

**Critério de Saída:** Para cada fator identificado deverá existir observações sobre como o fator influencia os resultados, como identificar a ocorrência da influência e como tratar esta influência na análise.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Estabelecer Critérios para Seleção do Projeto

## 3. Elaborar Plano do Piloto

Macroatividade que finaliza a etapa de planejamento e disponibiliza ao usuário o plano e as diretrizes para realização do piloto.

### 3.1. Elaborar Plano para Realização do Piloto

**Descrição:** Visualizar o plano e as diretrizes para realização do piloto. O plano para realização do piloto contém a concatenação de todos os artefatos gerados até esta atividade (identificação do piloto, ações de melhoria, plano de medição, expectativas da avaliação, critérios para seleção dos projetos-piloto e a lista de fatores de influência). As diretrizes para realização do piloto contém informações, um passo-a-passo, que irão orientar o gerente de projeto a planejar e executar o projeto para que as ações de melhoria sejam avaliadas durante a execução do(s) projeto(s) que servirá(ão) de piloto.

**Artefatos Requeridos:** Identificação do piloto, lista das ações de melhoria que serão testadas nos projetos, plano de medição para avaliação da proposta de melhoria, expectativas da avaliação, critérios para seleção do projeto, lista de fatores que podem influenciar nos resultados da avaliação.

**Artefatos Produzidos:** Plano do piloto, diretrizes para realização do piloto.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Estabelecer Critérios para Seleção do Projeto

## 4. Preparar para Executar Projeto-Piloto

Consiste na associação do projeto ao piloto e na preparação para realização do projeto-piloto conforme as diretrizes para realização do piloto.

### 4.1. Associar Projetos ao Piloto

**Descrição:** Formalizar a associação dos projetos ao piloto, caracterizando-os como projetos-piloto.

**Artefatos Requeridos:** Diretrizes para realização do piloto.

**Critérios de Entrada:** Os critérios estabelecidos para seleção dos projetos-piloto devem ser atendidos pelos projetos.

**Artefatos Produzidos:** Registro de comprometimento com a execução do piloto

**Crterios de Saída:** O registro de comprometimento deve explicitar se o projeto atende integralmente ou parcialmente os critrios definidos para seleo do projeto-piloto. Caso o atendimento seja parcial é necessrio interagir com o Grupo de Processo para verificar se há algum impedimento à associao do projeto ao piloto.

**Responsável:** Gerente do Projeto

**Participantes:** Gerente do Projeto

**Pré-atividade:** Visualizar Plano para Realizao do Piloto

#### **4.2. Preparar para execuo**

**Descrio:** Planejar o plano do processo para o projeto de forma a contemplar as aes de melhoria a serem testadas, preparar a equipe para realizar as aes de melhoria, incluir no planejamento do projeto todos os recursos necessrios para garantir a execuo adequada das aes de melhoria e preparar o ambiente para execuo do projeto-piloto.

**Artefatos Requeridos:** Diretrizes para realizao do piloto.

**Crterios de Entrada:** Os critrios estabelecidos para seleo dos projetos-piloto devem ser atendidos pelos projetos.

**Artefatos Produzidos:** Plano do processo para o projeto, plano/registo de treinamento (se pertinente) e plano do projeto.

**Crterios de Saída:** O plano do processo para o projeto deve contemplar todas as aes de melhoria prescritas nas diretrizes.

**Responsável:** Gerente do Projeto

**Participantes:** Gerente do Projeto

**Pré-atividade:** Visualizar Plano para Realizao do Piloto

### **5. Executar projetos-piloto**

A equipe do projeto executa as atividades previstas no plano do processo, inclusive as que constituem a proposta de melhoria. Em seguida, ou concomitantemente, as medidas são coletadas e depositadas na base de medies.

#### **5.1. Executar Proposta de Melhoria**

**Descrio:** Executar as aes de melhoria descritas na proposta de melhoria e que constam no plano do processo para o projeto.

**Artefatos Requeridos:** Plano do processo para o projeto

**Responsável:** Equipe do projeto

**Participantes:** Equipe do projeto

**Pré-atividade:** Preparar para execuo

#### **5.2. Coletar medidas**

**Descrio:** Coletar todas as medidas relacionadas às métricas contidas no plano de medioo do projeto (inclui todas as métricas listadas nas diretrizes para realizao do piloto).

**Artefatos Requeridos:** Plano de medioo do projeto-piloto

**Artefatos Produzidos:** Medidas para a base de medioo

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Executar Proposta de Melhoria

### **6. Monitorar Piloto**

Os projetos-piloto são monitorados com o intuito de acompanhar o progresso com relao à execuo das aes de melhoria e das atividades de coleta das medidas associadas às métricas definidas no plano de medioo do piloto. Reuniões de avaliao são realizadas para capturar as opinies dos participantes dos projetos-pilotos e identificar os fatores que estão afetando ou afetaram em algum momento a execuo da proposta de melhoria.

#### **6.1. Monitorar Projetos-Piloto**

**Descrio:** Verificar se o plano do processo para o projeto está sendo seguido e observar a situao das aes de melhoria durante a execuo do projeto. Durante a monitorao é interessante observar o comportamento da equipe e registrar as

impressões relacionadas à execução das ações de melhoria (dificuldades encontradas, resistência dos participantes etc.).

**Artefatos Requeridos:** Plano do processo para o projeto

**Crítérios de Entrada:** As ações de melhoria do plano do piloto devem constar no plano do processo para o projeto.

**Artefatos Produzidos:** Observações relevantes sobre a execução do projeto-piloto

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Preparar para execução

### **6.2. Conferir Medições dos Projetos-Piloto**

**Descrição:** Verificar se todas as medições estão sendo realizadas e se estão de acordo com os procedimentos de coleta para cada métrica contida no plano de medição.

**Artefatos Requeridos:** Plano do piloto (contém o plano de medição para avaliação da proposta de melhoria)

**Artefatos Produzidos:** Observações relevantes sobre a coleta das medidas (se pertinente).

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Monitorar Projetos-Piloto

### **6.3. Realizar Reunião de Avaliação**

**Descrição:** Obter uma avaliação da equipe quanto à utilidade das ações de melhoria para o projeto e registrar outros fatores que exercem influência na realização das atividades do projeto e no resultado das medições. Esta reunião pode ser executada ao final da execução das ações de melhoria, ao final do projeto e/ou sempre que for interessante coletar informações sobre a utilidade da proposta de melhoria.

**Artefatos Requeridos:** Lista de fatores de influência

**Artefatos Produzidos:** Avaliação da equipe do projeto.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo, Equipe do Projeto

**Pré-atividade:** Preparar para execução

## **7. Analisar Resultados dos Projetos-Piloto**

A macroatividade visa computar e comparar as medidas coletadas com as expectativas declaradas no plano do piloto para verificar quantitativamente o resultado obtido com a realização do piloto e realizar análise qualitativa considerando os fatores que influenciaram os resultados e a opinião dos participantes dos projetos-piloto.

### **7.1. Ratificar Fatores de Influência**

**Descrição:** Identificar os fatores que influenciaram os resultados dos projetos e como anular as influências na análise dos resultados.

**Artefatos Requeridos:** Lista de fatores de influência, avaliação da equipe do projeto.

**Artefatos Produzidos:** Lista de fatores que influenciaram os resultados

**Crítérios de Saída:** Todos os fatores devem ter sido analisados

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Realizar reunião de avaliação

### **7.2. Realizar Análise Quantitativa**

**Descrição:** Agregar os dados e analisar o resultado quantitativo obtido em cada projeto-piloto com relação às expectativas.

**Artefatos Requeridos:** Base de medições, plano do piloto (contém as expectativas da avaliação).

**Crítérios de Entrada:** Todos os projetos-piloto devem ter executado as ações descritas nas propostas de melhoria. A conferência da medição deve ter sido feita para todos os projetos-piloto.

**Artefatos Produzidos:** Análise quantitativa do piloto

**Critérios de Saída:** Um parecer preliminar da avaliação deve ser emitido considerando apenas os dados quantitativos.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Conferir Medições dos Projetos-Piloto

### **7.3. Realizar Análise Qualitativa**

**Descrição:** Emitir um parecer final a respeito da avaliação considerando o resultado da análise quantitativa, os fatores que influenciaram os resultados e a avaliação da equipe do projeto.

**Artefatos Requeridos:** Plano do piloto, análise quantitativa do piloto, lista de fatores que influenciaram os resultados e avaliação da equipe do projeto.

**Artefatos Produzidos:** Análise qualitativa do piloto.

**Critérios de Saída:** A análise deve apresentar um parecer sobre a efetividade da proposta de melhoria com relação à expectativa geral declarada para o piloto.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Realizar Análise Quantitativa

## **8. Finalizar Piloto**

A finalização do piloto consiste na elaboração do laudo contendo os resultados obtidos com o piloto, reunião dos artefatos incorporação à base de ativos e no registro das lições aprendidas.

### **8.1. Elaborar Laudo de Avaliação**

**Descrição:** Concatenar e formatar os dados em um único documento contendo o parecer a respeito da avaliação da efetividade da proposta de melhoria.

**Artefatos Requeridos:** Análise quantitativa e análise qualitativa do piloto

**Artefatos Produzidos:** Laudo de avaliação do piloto

**Critérios de Saída:** O laudo deve indicar o parecer da avaliação e os dados que balizaram a decisão.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Realizar Análise Qualitativa

### **8.2. Registrar Lições Aprendidas**

**Descrição:** Agregar os dados e analisar o resultado quantitativo obtido em cada projeto-piloto com relação às expectativas.

**Artefatos Requeridos:** Plano do piloto, análise quantitativa do piloto, lista de fatores que influenciaram os resultados, avaliação da equipe do projeto, análise qualitativa e laudo de avaliação.

**Critérios de Entrada:** Laudo com o parecer final do piloto

**Artefatos Produzidos:** Registro de lições aprendidas e ativos de processo.

**Critérios de Saída:** Lições aprendidas registradas e artefatos incorporados à base de ativos da organização.

**Responsável:** Grupo de Processo

**Participantes:** Grupo de Processo

**Pré-atividade:** Elaborar Laudo de Avaliação

## **ANEXO I - Notação para Modelagem de Processos**

A linguagem para modelagem de processos organizacionais foi proposta por VILLELA (2004) e é composta de elementos gráficos que podem ser do tipo área (Tabela A1.3), objeto (Tabela A1.1) ou ligação (Tabela A1.4), onde uma ligação estabelece uma relação entre dois objetos e uma área agrupa objetos, definindo um contexto para os mesmos. Objetos ainda permitem adornos (Tabela A1.2), utilizados para representar explicitamente características dos objetos. A seguir, cada elemento da linguagem é brevemente apresentado.

Tabela A1.1 – Definição e Notação dos Objetos

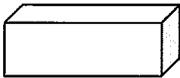
<b>Objeto</b>	<b>Notação</b>	<b>Definição</b>
Processo		Objeto referente ao conceito de mesmo nome definido na ontologia de organização (seção 5.7). <i>Atributos Especiais:</i> Origem (Interno, Externo)
Evento		Objeto que representa um acontecimento no ambiente que provoca o início ou fim de um processo. A notação é proveniente do produto comercial de <i>workflow</i> ARIS <i>ToolSet</i> .
Ator		Objeto que representa um pessoa, agente ou unidade organizacional. Estes conceitos encontram-se definidos na ontologia de organização. A notação foi utilizada para representação dos <i>workflows</i> básicos do <i>Rational Unified Process</i> .
Atividade		Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação foi utilizada para representação dos <i>workflows</i> básicos do <i>Rational Unified Process</i> . <i>Atributos Especiais:</i> Origem (Interna, Externa) Granularidade (Elementar ou Composta)
Estado Inicial		Objeto puramente notacional, proveniente dos diagramas de estado e que indica onde é iniciado o fluxo de atividades que definem um processo ou uma atividade composta
Estado Final		Objeto puramente notacional, proveniente dos diagramas de estado e que indica onde é encerrado o fluxo de atividades que definem um processo ou uma atividade composta
Conhecimento Explícito		Objeto que representa um conhecimento que pode ser expresso em palavras e números e ser facilmente transmitido e compartilhado.
Conhecimento Implícito		Objeto que representa um conhecimento que é altamente pessoal e difícil de formalizar, o que o torna também difícil de ser compartilhado.
Comunicação		Objeto que representa a comunicação de dados ou informações a partir da, ou para a, execução de uma atividade. A comunicação pode ser verbal ou escrita e exemplos são e-mail e fax.

Tabela A1.1 – Definição e Notação dos Objetos (continuação)

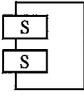
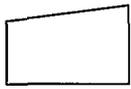
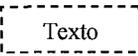
Objeto	Notação	Definição
Repositório (Meio Magnético)		Objeto que representa um meio magnético para o armazenamento de dados e informações. A notação é proveniente do produto comercial de <i>workflow ARIS ToolSet</i> .
Arquivo (Local Físico)		Objeto que representa um local físico para armazenamento de documentos e comunicações escritas.
Documento		Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação é proveniente do produto comercial de <i>workflow ARIS ToolSet</i> .
Componente de Hardware		Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação é baseada na notação de componente da UML.
Componente de Software		Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação é baseada na notação de componente da UML.
Peça		Objeto referente ao conceito de mesmo nome definido na ontologia de organização.
Matéria-Prima		Objeto referente ao conceito de mesmo nome definido na ontologia de organização.
Bem		Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação fornecida pode ser substituída por uma mais significativa para o objeto específico do modelo como, por exemplo, o logotipo do software. <i>Atributos Especiais:</i> Tipo (Usufruto, Software, Hardware e Equipamento de Produção)
Nota Explicativa		Objeto que permite que notas explicativas sejam adicionadas ao modelo. <i>Atributos Especiais:</i> Texto

Tabela A1.2 – Definição e Notação dos Adornos

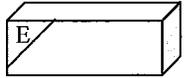
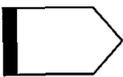
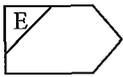
Objeto	Notação com Adornos	Definição dos Adornos
Processo		Adorno que indica que o processo é externo, ou seja, que é executado por outra organização.
Atividade	  (a) (b)	(a) Adorno que indica que a atividade é composta, o que significa que ela pode ser decomposta em sub-atividades; (b) Adorno que indica que a atividade é externa, ou seja, que é executada por outra organização.
Operação Lógica	   (a) (b) (c)	(a) Adorno que indica a operação lógica E; (b) Adorno que indica a operação lógica OU; (c) Adorno que indica a operação lógica OU Exclusivo.

Tabela A1.3 – Definição e Notação das Áreas

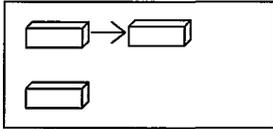
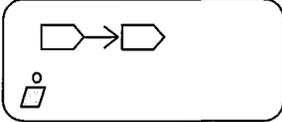
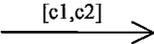
Objeto	Notação	Definição
Grupo de Processos		Área que agrupa processos relacionados.
Área de Ator		Área que agrupa atividades executadas por um ator ou grupo de atores. O ator ou o grupo de atores também precisa estar contido na área.

Tabela A1.4 – Definição e Notação das Ligações

Objeto	Notação	Definição
Fluxo de Controle		Ligação que indica a passagem de controle do objeto origem para o objeto destino. O c1 e o c2 indicados na notação são os rótulos das condições estabelecidas para que a passagem de controle ocorra. <i>Atributo Especial:</i> Condição, formada por rótulo e descrição
Fluxo de Entrada/Saída		Ligação que estabelece um insumo (se o fluxo é de entrada) ou um produto de uma atividade (se o fluxo é de saída). Quando o objeto de origem ou destino é um armazenador (repositório ou arquivo), a notação pode incluir os rótulos das informações trafegadas, existindo, então, um atributo especial. <i>Atributo Especial:</i> Informação, formada por rótulo e descrição
Ligação Não Direcionada		Ligação que não indica passagem de controle nem estabelece insumos e produtos para uma atividade, sendo utilizada para conectar bens de produção (software, hardware e equipamentos) utilizados como recursos para execução das atividades e para conectar eventos que atuam sobre processos, provocando o seu início ou fim. No segundo caso, um atributo especial é definido. <i>Atributo Especial:</i> Papel do Evento (Iniciador, Terminador)
Ligação para Nota Explicativa		Ligação que estabelece que uma nota explicativa é referente a um elemento do modelo.

# ANEXO II – Plano para realização do piloto elaborado pela BL Informática

---



## Plano para Realização do Piloto Avaliação da utilização do processo de 4 fases em projetos pequenos

### 1. IDENTIFICAÇÃO DO PILOTO

**Nome do Piloto:** Avaliação da utilização do processo de 4 fases em projetos pequenos

**Situação:** Não executado

**Propósito:** Verificar se a redução do número de fases do processo de desenvolvimento propicia a redução do custo da qualidade em projetos pequenos.

**Processo:** Processo Padrão

**Expectativa Geral:** O custo final do projeto não ultrapassar seis por cento do estimado, conforme previsto nos critérios orçamentários definidos na política organizacional. Espera-se reduzir em dez por cento o custo da qualidade (esforço do GC e do GQPP).

**Sugestão do Número de Projetos para Realizar o Piloto:** 1

**Número de Projetos Associados:** 0

### 2. MELHORIA A SER AVALIADA

1. **Ação de melhoria:** Redução do número de fases do processo de desenvolvimento de sete para quatro fases. A Elicitação, Análise e Projeto do Sistema (fase 2) e atividades complementares de planejamento da fase 3 foram antecipadas para fase 1; As atividades de preparação para integração e integração do software e sistema (fases 5 e 6) foram concatenadas em uma única fase.

### 3. PLANO DE MEDIÇÃO

**Objetivo: Diminuir o Custo da Qualidade (Esforço GC e GQPP)**

**Questão: Qual o custo da qualidade (esforço) do GC?**

1. **Métrica:** Esforço do GC

**Mnemônico:** EGC

**Atomicidade:** Atômica

**Unidade de medida:** horas

**Valor base:** 0.00

**Procedimento análise:** Análise ao final do projeto

Descrição: Analisar ao final do projeto se o custo da qualidade esforço do GC diminuir em relação aos projetos anteriores.

Forma apresentação: Gráfico de barras

Formato apresentação data: m/Y

**Procedimento coleta:** MANUAL

Descrição: A Coleta é feita manualmente ao final do projeto pelo Gerente do Projeto

Tipo: Manual

Tipo Entidade: projeto

**Questão: Qual o custo da qualidade (esforço) do GQPP?**

2. **Métrica:**Esforço do GQPP - Custo da qualidade do GC com relação ao esforço demandado pelas atividades de gerência de configuração  
**Mnemônico:**EGQPP  
**Atomicidade:**Atômica  
**Unidade de medida:**horas  
**Valor base:**0.00  
**Procedimento análise:**Análise Mensal  
Descrição:Analisar ao final do projeto se o custo da qualidade esforço do GQPP diminuir em relação aos projetos anteriores.  
Forma apresentação:Gráfico de barras  
Formato apresentação data:m/Y  
**Procedimento coleta:**MANUAL  
Descrição:A coleta é feita manualmente pelo gerente do projeto ao final do projeto  
Tipo:Manual  
Tipo Entidade:projeto

**Questão: Qual o percentual de redução do esforço do GC?**

3. **Métrica:**Percentual do esforço do GC - Custo da qualidade do GC com relação ao esforço demandado pelas atividades de gerência de configuração  
**Mnemônico:**PEGC  
**Atomicidade:**Não atômica  
**Equação cálculo:**EGC/ERP  
**Unidade de medida:**%  
**Valor base:**0.00  
**Composição da métrica:**  
1. Esforço real do projeto (ERP)  
**Procedimento análise:**Análise ao final do projeto  
Descrição:Ao final do projeto será analisado o percentual de esforço do GC em relação ao esforço real do projeto.  
Forma apresentação:Gráfico de barras  
Formato apresentação data:m/Y  
**Procedimento coleta:**MANUAL  
Descrição:Ao final do projeto o Gerente de projeto é responsável por coletar o percentual do esforço do GC.  
Observação:  
Tipo:Manual  
Tipo Entidade:projeto

**Questão: Qual o percentual de redução do esforço do GQPP?**

4. **Métrica:**Percentual de esforço do GQPP -  
**Mnemônico:**PEGQPP  
**Atomicidade:**Atômica

**Equação cálculo:**EGQPP/ERP

**Unidade de medida:**%

**Valor base:**5.00

**Composição da métrica:**

1. Esforço real do projeto (ERP)

**Procedimento análise:**Análise ao final do projeto

**Descrição:**Ao final do projeto será analisado o percentual de esforço do GC em relação ao esforço real do projeto.

**Forma apresentação:**Gráfico de barras

**Formato apresentação data:**m/Y

**Procedimento coleta:**AUTOMATICO

**Tipo:**Automático do repositório do projeto

**Tipo Entidade:**projeto

### **Objetivo: Monitorar o processo Gerência de Requisitos**

**Questão: Qual a estabilidade dos requisitos após sua aprovação?**

5. **Métrica:**Número de Requisitos Alterados

**Descrição:**Número de requisitos aprovados que sofreram alterações ao longo do projeto.

**Mnemônico:**NRA

**Atomicidade:**Atômica

**Forma apresentação:**Gráfico de barras

**Formato apresentação data:**m/Y

**Procedimento coleta:**Manual

**Observação:**Esta métrica só pode ser coletada após aprovação dos requisitos na fase 2

**Tipo:**Manual

**Tipo Entidade:**projeto

6. **Métrica:**Número de Requisitos Novos

**Descrição:**número de novos requisitos após a aprovação na fase 2.

**Mnemônico:**NRN

**Atomicidade:**Atômica

**Forma apresentação:**Gráfico de barras

**Formato apresentação data:**m/Y

**Procedimento coleta:**Manual

**Descrição:**

**Observação:**Esta métrica só pode ser coletada após aprovação dos requisitos na fase 2

**Tipo:**Manual

**Tipo Entidade:**projeto

7. **Métrica:**Número de Requisitos Excluídos

**Descrição:**número de requisitos que foram excluídos após a aprovação na fase 2.

**Mnemônico:**NRE

**Atomicidade:**Atômica

**Unidade de medida:**

**Valor base:**0.00

**Procedimento análise:**

**Descrição:**

**Forma apresentação:**Gráfico de barras

**Formato apresentação data:**m/Y

**Procedimento coleta:**Manual

Descrição:

Observação:Esta métrica só pode ser coletada após aprovação dos requisitos na fase 2

Tipo:Manual

Tipo Entidade:projeto

8. **Métrica:**Estabilidade dos requisitos aprovados (ERA)

**Descrição:**Apresenta o número de alterações ocorridas nos requisitos após sua aprovação. É a soma do número de requisitos alterados após a aprovação, número de requisitos adicionados e número de requisitos excluídos.

**Mnemônico:**ERA

**Atomicidade:**Não atômica

**Equação cálculo:**NRA+NRN+NRE

**Unidade de medida:**

**Valor base:**0.00

**Composição da métrica:**

1. Número de Requisitos Alterados
2. Número de Requisitos Novos
3. Número de Requisitos Excluídos

**Procedimento análise:**Análise Mensal

Descrição:após a aprovação dos requisitos, mensalmente será contado o número de vezes que um requisito aprovados sofreu alterações, o número de requisitos novos e o número de requisitos excluídos. A soma desses três valores indicará a estabilidade dos requisitos no mês. O valor obtido para a estabilidade dos requisitos será, então, comparado com os valores dos meses anteriores. A meta é diminuir o número de alterações de requisitos ao longo do projeto, caso o valor atual seja maior ou igual ao valor anterior, abrir uma RNACP para analisar as causas e criar ações corretivas e/ou preventivas.

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Automático

Descrição:As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.

Observação:Responsável: Líder do Projeto; Quando coletar: mensalmente, após a aprovação dos requisitos na fase 2.

Tipo:Automático do repositório do projeto

Tipo Entidade:projeto

**Objetivo: Monitorar o processo Monitoração e Controle do Projeto**

**Questão: Com relação ao trabalho já executado e seu custo, quanto devo gastar?**

9. **Métrica:**BAC

**Descrição:**Custo Previsto para o Projeto

**Mnemônico:**BAC

**Atomicidade:**Atômica

**Unidade de medida:**R\$

**Valor base:**0.00

**Procedimento análise:**

Descrição:

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Coletar do repositório do projeto no SGP

10. **Métrica:**CPI

**Descrição:**Índice de desempenho de Custo

**Mnemônico:**CPI

**Atomicidade:**Não atômica

**Equação cálculo:**EV/AC

**Unidade de medida:**R\$

**Valor base:**0.00

**Composição da métrica:**

1. Valor Orçado para o Trabalho Realizado
2. AC

**Procedimento análise:**Análise Mensal

**Descrição:**Mensalmente será analisada o índice de desempenho de custo do projeto no momento da confecção do 2º Relatório de Monitoração e Controle. Este índice deve estar entre 0,85 e 1,15. Se a métrica ficar fora da meta, abrir uma RNACP para analisar as causas e criar ações corretivas e/ou preventivas.

**Forma apresentação:**Gráfico de barras

**Formato apresentação data:**m/Y

**Procedimento coleta:**Coletar do repositório do projeto no SGP

**Descrição:**As métricas são derivadas automaticamente a partir de valores registrados no SGP e atualizadas manualmente no Ambiente TABA.

**Observação:**

**Tipo:**Automático do repositório do projeto

**Tipo Entidade:**projeto

**Atributo Entidade:**

**Agrupar por:**

**Tipo Entidade Agregada:**

**Regras:**

11. **Métrica:**EAC

**Descrição:**Estimativa de Custo ao final do projeto

**Mnemônico:**EAC

**Atomicidade:**Atômica

**Equação cálculo:**BAC/CPI

**Unidade de medida:**

**Valor base:**0.00

**Composição da métrica:**

1. BAC
2. CPI

**Procedimento análise:**Análise Mensal

**Descrição:**mensalmente será analisada o custo estimado para conclusão do projeto. O custo estimado para conclusão do projeto deve ser comparado com o tempo e custo planejado do projeto segundo a última baseline aprovada pelo cliente. O custo estimado pode ser no máximo 15% maior ou menor que o custo planejado. Se a métrica ficar fora da meta, abrir uma RNACP para analisar as causas e criar ações corretivas e/ou preventivas.

**Forma apresentação:**Gráfico de barras

**Formato apresentação data:**m/Y

**Procedimento coleta:**Coletar do repositório do projeto no SGP

**Descrição:**As métricas são derivadas automaticamente a partir de valores registrados no SGP e atualizadas manualmente no Ambiente TABA.

**Observação:**

Tipo:Automático do repositório do projeto  
Tipo Entidade:projeto

12. **Métrica:AC**

**Descrição:**Custo Atual do Projeto

**Mnemônico:**AC

**Atomicidade:**Atômica

**Unidade de medida:**

**Valor base:**0.00

**Procedimento análise:**

Descrição:

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Coletar do Repositório do projeto no SGP

13. **Métrica:CV**

**Mnemônico:**CV

**Atomicidade:**Não atômica

**Equação cálculo:**EV-AC

**Unidade de medida:**R\$

**Valor base:**0.00

**Composição da métrica:**

1. Valor Orçado para o Trabalho Realizado
2. AC

**Procedimento análise:**Análise Mensal

Descrição:mensalmente será analisada se custo do projeto foi ultrapassado.

Se a variação do custo for negativa o projeto está com os custos ultrapassados. Deve ser criada uma RNACP somente se o índice de custo (CPI) estiver fora da meta.

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Coletar do repositório do projeto no SGP

Descrição:As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.

Observação:Responsável: Gerente do Projeto. Quando coletar: quinzenalmente

Tipo:Automático do repositório do projeto

Tipo Entidade:projeto

14. **Métrica:EACTC**

**Mnemônico:**EACTC

**Atomicidade:**Não atômica

**Equação cálculo:** $(BAC-AC)/CPI+AC$

**Unidade de medida:**R\$

**Valor base:**0.00

**Composição da métrica:**

1. BAC
2. AC
3. CPI

**Procedimento análise:**Análise Mensal

Descrição:mensalmente será analisada se é válido para empresa aplicar ações corretivas. Caso o custo total do projeto ganho com as ações corretivas seja

menor ou igual ao custo das ações corretivas é melhor manter o índice de performance atual. A RNACP deve ser aberta caso o procedimento de análise do EACTC não tenha atingido sua meta.

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Coletar do repositório do projeto no sgp

Descrição:As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.

Observação:

Tipo:Manual

Tipo Entidade:projeto

#### 15. **Métrica:**Valor Orçado para o Trabalho Realizado

**Descrição:**É o valor resultante percentual realizado de todas as solicitações cadastradas no SGP (PRTT) aplicado ao custo total do projeto (BAC).

**Mnemônico:**EV

**Atomicidade:**Não atômica

**Equação cálculo:**PRTT\*BAC

**Unidade de medida:**R\$

**Valor base:**0.00

**Composição da métrica:**

1. PRTT
2. BAC

**Procedimento análise:**

Descrição:

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**

Descrição:As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.

Observação:

Tipo:Automático do repositório do projeto

Tipo Entidade:projeto

Atributo Entidade:

Agrupar por:

Tipo Entidade Agregada:

Regras:

#### 4. **EXPECTATIVAS DA AVALIAÇÃO**

##### 1. **Percentual do esforço do GC**

**Expectativa:**Menor ou igual à 5 (%)

**Justificativa:**Atualmente foi observado que o esforço do GC tem superado os 5verificados nos processos de nível 2, com a inclusao do processos de nível 3. Foram realizadas melhorias neste processo, reduzindo o numero de fases sem excluir a quantidade de artefatos verificados. A expectativa e que este valor fique menor ou igual ao constatado no processo de nível 2.

##### 2. **Percentual de esforço do GQPP**

**Expectativa:**Menor ou igual à 10 (%)

**Justificativa:**Atualmente foi observado que o esforço do GQPP tem superado os 5verificados nos processos de nível 2, com a inclusao do processos de nível 3.

Foram realizadas melhorias neste processo, reduzindo o numero de fases sem excluir a quantidade de artefatos verificados. A expectativa e que este valor fique menor ou igual ao constatado a 10, ja que a organizacao, considera aceitavel o teto de 15 % do esforco do projeto para as atividades de qualidade do GC e do GQPP.

**3. Percentual de excedido para o custo do projeto da ultima baseline**

**Expectativa:**Menor ou igual à 6 (%)

**Justificativa:**O criterio orcamentário estabelecido pela organizacao de 6 % deve também ser possível nos projetos pequenos utilizando este processo de 4 fases.

**5. CRITÉRIOS PARA SELEÇÃO DO PROJETO**

**Paradigma:** Estruturado; Orientado a Objetos;

**Tamanho do Projeto:** Pequeno (<2500 homem/hora);

**Complexidade do Projeto:** Baixa; Média;

**Nível de Experiência da Gerência:** Médio;

**Experiência da Equipe em Engenharia de Software:** Média;

**Experiência da Equipe com a Tecnologia:** Média;

**6. FATORES QUE PODEM INFLUENCIAR A AVALIAÇÃO**

**1. Equipe de desenvolvimento indisponível**

**Como identificar:**Se as pessoas previstas inicialmente foram substituidas.

**Como influencia:**Caso a pessoa alocada fique indisponivel e a outra pessoa precise ser treinada ou auxiliada podem implicar no aumento do esforco do projeto e do custo da qualidade.

**Como tratar:**Contratar profissional externo com as mesmas habilidades para realizar as tarefas afetadas.

# ANEXO III – Diretrizes para realização do piloto na BL Informática



## Diretrizes para Realização do Piloto 'Avaliação da utilização do processo de 4 fases em projetos pequenos'

O objetivo deste documento é fazer com que as ações de melhoria que estão sendo avaliadas sejam aplicadas adequadamente no projeto. Seguir as diretrizes é imprescindível para fazer com as avaliações das ações de melhoria sejam realizadas corretamente.

**Caso o projeto esteja sendo utilizado para realizar mais de um piloto, siga as diretrizes em paralelo, executando o passo 1 em todas as diretrizes e depois o passo 2 e assim sucessivamente.**

Siga os passos a seguir:

### 1. VERIFIQUE SE O SEU PROJETO CORRESPONDE AOS CRITÉRIOS DO PILOTO

Com base nas informações que você possui sobre o projeto, verifique se ele corresponde aos critérios listados a seguir:

**Paradigma:** Estruturado; Orientado a Objetos;

**Tamanho do Projeto:** Pequeno (<2500 homem/hora);

**Complexidade do Projeto:** Baixa; Média;

**Nível de Experiência da Gerência:** Médio;

**Experiência da Equipe em Engenharia de Software:** Média;

**Experiência da Equipe com a Tecnologia:** Média;

Se o seu projeto NÃO corresponde aos critérios, repasse esta informação ao grupo de processo. Do contrário, continue seguindo as diretrizes.

### 2. ASSOCIE O SEU PROJETO AO PILOTO

Ao informar o nome do projeto e o escopo, na tela de cadastro de projetos no AdaptPro, clique na opção 'Associar Piloto' e selecione o piloto **Avaliação da utilização do processo de 4 fases em projetos pequenos**

### 3. VERIFIQUE AS ALTERAÇÕES NO PROCESSO PARA O PROJETO

Certifique-se que o processo para o projeto será contemplado com as alterações descritas a seguir:

1. **Ação de melhoria:** Redução do número de fases do processo de desenvolvimento de sete para quatro fases. A Elicitação, Análise e Projeto do Sistema (fase 2) e atividades complementares de planejamento da fase 3 foram antecipadas para fase 1; As atividades de preparação para integração e integração do software e sistema (fases 5 e 6) foram concatenadas em uma única fase.

Caso seja necessário, realize a adaptação do processo para o projeto na ferramenta AdaptPro, na atividade 'Planejar Processo > Incluir/Excluir Atividades'.

### 4. TREINE A EQUIPE

Analisando as ações de melhoria, verifique os conhecimentos e habilidades necessárias para garantir que as ações de melhoria sejam executadas de forma adequada. Caso seja pertinente, planeje e execute o treinamento necessário para equipe.

## 5. PLANEJE E GERENCIE OS RISCOS AO PROJETO

Além dos riscos inerentes a execução do projeto, outros fatos podem influenciar nos resultados do projeto e, conseqüentemente, a avaliação das ações de melhoria. Portanto, trate os fatos a seguir como riscos ao projeto incluindo-os no plano de riscos e gerenciando-os a medida que o projeto é executado.

### 1. Equipe de desenvolvimento indisponível

**Como identificar:** Se as pessoas previstas inicialmente foram substituídas.

**Como influencia:** Caso a pessoa alocada fique indisponível e a outra pessoa precise ser treinada ou auxiliada podem implicar no aumento do esforço do projeto e do custo da qualidade.

**Como tratar:** Contratar profissional externo com as mesmas habilidades para realizar as tarefas afetadas.

## 6. INCLUA MÉTRICAS NO PLANO DE MEDIÇÃO NO PROJETO

O plano de medição apresentado a seguir deve integrar o plano de medição do projeto. A inclusão destas métricas no plano de medição e a coleta adequada das medidas são determinantes para o sucesso da avaliação.

**Objetivo: Diminuir o Custo da Qualidade (Esforço GC e GQPP)**

**Questão: Qual o custo da qualidade (esforço) do GC?**

### 2. Métrica: Esforço do GC

**Mnemônico:** EGC

**Atomicidade:** Atômica

**Unidade de medida:** horas

**Valor base:** 0.00

**Procedimento análise:** Análise ao final do projeto

Descrição: Analisar ao final do projeto se o custo da qualidade esforço do GC diminuir em relação aos projetos anteriores.

Forma apresentação: Gráfico de barras

Formato apresentação data: m/Y

**Procedimento coleta:** MANUAL

Descrição: A coleta é feita manualmente ao final do projeto pelo Gerente do Projeto

Tipo: Manual

Tipo Entidade: projeto

**Questão: Qual o custo da qualidade (esforço) do GQPP?**

### 3. Métrica: Esforço do GQPP - Custo da qualidade do GC com relação ao esforço demandado pelas atividades de gerência de configuração

**Mnemônico:** EGQPP

**Atomicidade:** Atômica

**Unidade de medida:**horas

**Valor base:**0.00

**Procedimento análise:**Análise Mensal

Descrição:Analisar ao final do projeto se o custo da qualidade esforço do GQPP diminuir em relação aos projetos anteriores.

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**MANUAL

Descrição:A coleta é feita manualmente pelo gerente do projeto ao final do projeto

Tipo:Manual

Tipo Entidade:projeto

#### **Questão: Qual o percentual de redução do esforço do GC?**

4. **Métrica:**Percentual do esforço do GC - Custo da qualidade do GC com relação ao esforço demandado pelas atividades de gestão de configuração

**Mnemônico:**PEGC

**Atomicidade:**Não atômica

**Equação cálculo:**EGC/ERP

**Unidade de medida:**%

**Valor base:**0.00

**Composição da métrica:**

1. Esforço real do projeto (ERP)

**Procedimento análise:**Análise ao final do projeto

Descrição: Ao final do projeto será analisado o percentual de esforço do GC em relação ao esforço real do projeto.

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**MANUAL

Descrição: Ao final do projeto o Gerente de projeto é responsável por coletar o percentual do esforço do GC.

Observação:

Tipo:Manual

Tipo Entidade:projeto

#### **Questão: Qual o percentual de redução do esforço do GQPP?**

5. **Métrica:**Percentual de esforço do GQPP -

**Mnemônico:**PEGQPP

**Atomicidade:**Atômica

**Equação cálculo:**EGQPP/ERP

**Unidade de medida:**%

**Valor base:**5.00

**Composição da métrica:**

1. Esforço real do projeto (ERP)

**Procedimento análise:**Análise ao final do projeto

Descrição: Ao final do projeto será analisado o percentual de esforço do GC em relação ao esforço real do projeto.

Forma apresentação:Gráfico de barras  
Formato apresentação data:m/Y  
**Procedimento coleta:**AUTOMATICO  
Tipo:Automático do repositório do projeto  
Tipo Entidade:projeto

### **Objetivo: Monitorar o processo Gerência de Requisitos**

#### **Questão: Qual a estabilidade dos requisitos após sua aprovação?**

6. **Métrica:**Número de Requisitos Alterados  
**Descrição:**Número de requisitos aprovados que sofreram alterações ao longo do projeto.  
**Mnemônico:**NRA  
**Atomicidade:**Atômica  
Forma apresentação:Gráfico de barras  
Formato apresentação data:m/Y  
**Procedimento coleta:**Manual  
Observação:Esta métrica só pode ser coletada após aprovação dos requisitos na fase 2  
Tipo:Manual  
Tipo Entidade:projeto
7. **Métrica:**Número de Requisitos Novos  
**Descrição:**número de novos requisitos após a aprovação na fase 2.  
**Mnemônico:**NRN  
**Atomicidade:**Atômica  
Forma apresentação:Gráfico de barras  
Formato apresentação data:m/Y  
**Procedimento coleta:**Manual  
Descrição:  
Observação:Esta métrica só pode ser coletada após aprovação dos requisitos na fase 2  
Tipo:Manual  
Tipo Entidade:projeto
8. **Métrica:**Número de Requisitos Excluídos  
**Descrição:**número de requisitos que foram excluídos após a aprovação na fase 2.  
**Mnemônico:**NRE  
**Atomicidade:**Atômica  
**Unidade de medida:**  
**Valor base:**0.00  
**Procedimento análise:**  
Descrição:  
Forma apresentação:Gráfico de barras  
Formato apresentação data:m/Y  
**Procedimento coleta:**Manual  
Descrição:  
Observação:Esta métrica só pode ser coletada após aprovação dos requisitos na fase 2  
Tipo:Manual  
Tipo Entidade:projeto
9. **Métrica:**Estabilidade dos requisitos aprovados (ERA)  
**Descrição:**Apresenta o número de alterações ocorridas nos requisitos após

sua aprovação. É a soma do número de requisitos alterados após a aprovação, número de requisitos adicionados e número de requisitos excluídos.

**Mnemônico:**ERA

**Atomicidade:**Não atômica

**Equação cálculo:** $NRA+NRN+NRE$

**Unidade de medida:**

**Valor base:**0.00

**Composição da métrica:**

1. Número de Requisitos Alterados
2. Número de Requisitos Novos
3. Número de Requisitos Excluídos

**Procedimento análise:**Análise Mensal

Descrição:após a aprovação dos requisitos, mensalmente será contado o número de vezes que um requisito aprovados sofreu alterações, o número de requisitos novos e o número de requisitos excluídos. A soma desses três valores indicará a estabilidade dos requisitos no mês. O valor obtido para a estabilidade dos requisitos será, então, comparado com os valores dos meses anteriores. A meta é diminuir o número de alterações de requisitos ao longo do projeto, caso o valor atual seja maior ou igual ao valor anterior, abrir uma RNACP para analisar as causas e criar ações corretivas e/ou preventivas.

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Automático

Descrição:As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.

Observação:Responsável: Líder do Projeto; Quando coletar: mensalmente, após a aprovação dos requisitos na fase 2.

Tipo:Automático do repositório do projeto

Tipo Entidade:projeto

## **Objetivo: Monitorar o processo Monitoração e Controle do Projeto**

**Questão: Com relação ao trabalho já executado e seu custo, quanto devo gastar?**

### 10. **Métrica:**BAC

**Descrição:**Custo Previsto para o Projeto

**Mnemônico:**BAC

**Atomicidade:**Atômica

**Unidade de medida:**R\$

**Valor base:**0.00

**Procedimento análise:**

Descrição:

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Coletar do repositório do projeto no SGP

### 11. **Métrica:**CPI

**Descrição:**Índice de desempenho de Custo

**Mnemônico:**CPI

**Atomicidade:**Não atômica

**Equação cálculo:**EV/AC

**Unidade de medida:**R\$

**Valor base:**0.00

**Composição da métrica:**

1. Valor Orçado para o Trabalho Realizado
2. AC

**Procedimento análise:**Análise Mensal

Descrição:Mensalmente será analisada o índice de desempenho de custo do projeto no momento da confecção do 2º Relatório de Monitoração e Controle. Este índice deve estar entre 0,85 e 1,15. Se a métrica ficar fora da meta, abrir uma RNACP para analisar as causas e criar ações corretivas e/ou preventivas.

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Coletar do repositório do projeto no SGP

Descrição:As métricas são derivadas automaticamente a partir de valores registrados no SGP e atualizadas manualmente no Ambiente TABA.

Observação:

Tipo:Automático do repositório do projeto

Tipo Entidade:projeto

Atributo Entidade:

Agrupar por:

Tipo Entidade Agregada:

Regras:

## 12. Métrica:EAC

**Descrição:**Estimativa de Custo ao final do projeto

**Mnemônico:**EAC

**Atomicidade:**Atômica

**Equação cálculo:**BAC/CPI

**Unidade de medida:**

**Valor base:**0.00

**Composição da métrica:**

1. BAC
2. CPI

**Procedimento análise:**Análise Mensal

Descrição:mensalmente será analisada o custo estimado para conclusão do projeto. O custo estimado para conclusão do projeto deve ser comparado com o tempo e custo planejado do projeto segundo a última baseline aprovada pelo cliente. O custo estimado pode ser no máximo 15maior ou menor que o custo planejado. Se a métrica ficar fora da meta, abrir uma RNACP para analisar as causas e criar ações corretivas e/ou preventivas.

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Coletar do repositório do projeto no SGP

Descrição:As métricas são derivadas automaticamente a partir de valores registrados no SGP e atualizadas manualmente no Ambiente TABA.

Observação:

Tipo:Automático do repositório do projeto

Tipo Entidade:projeto

## 13. Métrica:AC

**Descrição:**Custo Atual do Projeto

**Mnemônico:**AC

**Atomicidade:**Atômica

**Unidade de medida:**

**Valor base:**0.00

**Procedimento análise:**

Descrição:

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Coletar do Repositório do projeto no SGP

14. **Métrica:**CV

**Mnemônico:**CV

**Atomicidade:**Não atômica

**Equação cálculo:**EV-AC

**Unidade de medida:**R\$

**Valor base:**0.00

**Composição da métrica:**

1. Valor Orçado para o Trabalho Realizado
2. AC

**Procedimento análise:**Análise Mensal

Descrição:mensalmente será analisada se custo do projeto foi ultrapassado. Se a variação do custo for negativa o projeto está com os custos ultrapassados. Deve ser criada uma RNACP somente se o índice de custo (CPI) estiver fora da meta.

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Coletar do repositório do projeto no SGP

Descrição:As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.

Observação:Responsável: Gerente do Projeto. Quando coletar: quinzenalmente

Tipo:Automático do repositório do projeto

Tipo Entidade:projeto

15. **Métrica:**EACTC

**Mnemônico:**EACTC

**Atomicidade:**Não atômica

**Equação cálculo:** $(BAC-AC)/CPI+AC$

**Unidade de medida:**R\$

**Valor base:**0.00

**Composição da métrica:**

1. BAC
2. AC
3. CPI

**Procedimento análise:**Análise Mensal

Descrição:mensalmente será analisada se é válido para empresa aplicar ações corretivas. Caso o custo total do projeto ganho com as ações corretivas seja menor ou igual ao custo das ações corretivas é melhor manter o índice de performance atual. A RNACP deve ser aberta caso o procedimento de análise do EACTC não tenha atingido sua meta.

Forma apresentação:Gráfico de barras

Formato apresentação data:m/Y

**Procedimento coleta:**Coletar do repositório do projeto no sgp  
Descrição:As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.  
Observação:  
Tipo:Manual  
Tipo Entidade:projeto

16. **Métrica:**Valor Orçado para o Trabalho Realizado  
**Descrição:**É o valor resultante percentual realizado de todas as solicitações cadastradas no SGP (PRTT) aplicado ao custo total do projeto (BAC).  
**Mnemônico:**EV  
**Atomicidade:**Não atômica  
**Equação cálculo:**PRTT\*BAC  
**Unidade de medida:**R\$  
**Valor base:**0.00  
**Composição da métrica:**
- |    |      |
|----|------|
| 1. | PRTT |
| 2. | BAC  |

**Procedimento análise:**

Descrição:  
Forma apresentação:Gráfico de barras  
Formato apresentação data:m/Y

**Procedimento coleta:**

Descrição:As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.  
Observação:  
Tipo:Automático do repositório do projeto  
Tipo Entidade:projeto  
Atributo Entidade:  
Agrupar por:  
Tipo Entidade Agregada:  
Regras:

**7. MONITORE OS RISCOS E REALIZE A COLETA DAS MEDIDAS**

Durante a execução do projeto, não deixe de monitorar os riscos e realizar a coleta das medidas relativa às métricas contidas no plano de medição do projeto.

**8. MANTENHA O GRUPO DE PROCESSO INFORMADO**

A qualquer momento, caso surja algum fato, além daqueles previstos inicialmente, que pode influenciar as ações de melhoria, bem como, na coleta das medidas associadas às métricas contidas nas diretrizes, informe imediatamente ao grupo de processo.

# ANEXO IV – Registro de monitoração do piloto na BL Informática

---



BL Informática

Registro de Monitoração do Piloto

## Registro de Monitoração do Piloto para Avaliação da Utilização do Processo de 4 Fases em Projetos Pequenos

### 1. IDENTIFICAÇÃO DO PILOTO

**Nome do Piloto:** Avaliação da utilização do processo de 4 fases em projetos pequenos

**Situação:** Em execução – Fase 2 – Encerramento da Fase

**Propósito:** Verificar se a redução do número de fases do processo de desenvolvimento propicia a redução do custo da qualidade em projetos pequenos.

**Processo:** Processo Padrão

**Expectativa Geral:** Não ultrapassar os seis por cento previstos nos critérios orçamentários definidos na política organizacional. Reduzir em dez por cento o custo da qualidade (esforço do GC e do GQPP).

**Sugestão do Número de Projetos para Realizar o Piloto:** 1

**Número de Projetos Associados:** 1

### 2. REGISTROS DE MONITORAÇÃO

#### 2.1. Projeto-Piloto pControlWindows (BL 00177/2006)

**Data:** 28/04/2005

**Situação:** Em execução

**Responsável pelo registro:** Analia Irigoyen Ferreiro Ferreira

**Papel:** Gerente de Desenvolvimento

**Observações:**

Grupo de Gerência de Configuração – Não notou mudança significativa no tempo de trabalho, já que os documentos aumentaram mas houve uma redução significativa do número de versões e solicitações de mudanças analisadas.

Grupo do PPQA – Distribuição do tempo trabalhado. A fase ficou maior, facilitando a distribuição das suas avaliações.

Gerente do Projeto (Roberta da Silva Cerqueira) – A redução do número de fases facilitou o planejamento e distribuição das horas estimadas pelo APF.

**Data:**

**Situação:**

**Responsável pelo registro:** Analia Irigoyen Ferreiro Ferreira

**Papel:** Gerente de Desenvolvimento

**Observações:**