



Pós-Graduação em Ciência da Computação

“Uma Abordagem para Implantação de Testes baseada em Metodologias Ágeis”

Por

Juliana Ochner
Dissertação de Mestrado



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, FEVEREIRO/2008



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JULIANA OCHNER

Uma Abordagem para Implantação de Testes baseada em Metodologias Ágeis

*ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA
UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO
PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA
DA COMPUTAÇÃO.*

*ORIENTADOR(A): Prof. Dr. Alexandre Marcos Lins de
Vasconcelos.*

RECIFE, FEVEREIRO/2008

Para minha família e amigos. Amo muito vocês.

AGRADECIMENTOS

Agradeço a todos que contribuíram, direta ou indiretamente, para o desenvolvimento desta dissertação.

Ao meu orientador, Dr. Alexandre M. L. Vasconcelos, por ter me aceitado como sua aluna e com isso, ter tornado o meu estudo muito mais interessante.

À Renata Endriss por ter me introduzido à área de testes e por me aconselhar nas horas mais difíceis.

Ao time do projeto Siemens-EPM pela paciência e pelo apoio que me deram quando tinham que escutar os meus desabafos diários sobre o meu trabalho do mestrado.

A todos os meus amigos pela torcida.

À minha mãe que mesmo longe sempre me apoiou e me deu forças para continuar seguindo em frente. Ao meu pai que onde estiver tenho certeza que está olhando por mim. Ao meu irmãzinho pelo orgulho que ele sente da sua Tata.

Ao meu namorado, Roberto, pela paciência, compreensão e dedicação que teve comigo durante todos estes anos.

Agradeço a Deus, sem ele nada disso seria possível.

UMA ABORDAGEM PARA IMPLANTAÇÃO DE TESTES SOFTWARE BASEADA EM METODOLOGIAS ÁGEIS

RESUMO

Nas últimas décadas, indústrias de software têm investido um esforço substancial na melhoria da qualidade de seus produtos. Pensando nisso, muitas empresas estão buscando implantar ou melhorar seus processos de teste. A melhoria de processo de testes é uma instanciamento da melhoria de processos. Com o intuito de facilitar a melhoria de processo de software, diversos modelos têm surgido, tais como IDEAL [McFeeley, 1996], ISO/IEC 15504 [ISO 15504, 2005] e Pro2Pi [Salviano, 2006]. Existem também modelos de referência específicos para definição de requisitos necessários a um Processo de Testes, como: TMM [Burnstein, 2003], TIM [Ericson, 1996] e TPI [Koomen, 1999]. No entanto, estes modelos não fornecem diretrizes para a sua implantação em uma organização de testes. A melhoria de processo de testes é uma instanciamento da melhoria de processos.

Embora tenham surgido tais modelos de melhoria de processo, bem como os modelos de referência para processos de testes, e haja resultados encorajadores, grande parte da indústria de software ainda apresenta muitas dificuldades em realizar um programa de melhoria de processo de testes de software efetivo. Isto ocorre por uma série de motivos, dentre os quais, podemos destacar o fato de que estes modelos são “genéricos” e exigem diversas adaptações para tratar as dificuldades comumente encontradas em um programa de melhoria de testes. Diante deste cenário, a abordagem proposta neste trabalho guiará as empresas de maneira simples e ágil a estabelecer processos de teste de software, possibilitando o alcance de melhores níveis de qualidade de seus produtos.

Palavras-chave: Testes de Software, Modelo de Melhoria de Processo, Modelo de Maturidade em Testes.

AN APPROACH FOR DEPLOYING SOFTWARE TESTING BASED ON AGILE METHODOLOGIES

ABSTRACT

In the last years, software industry has invested a substantial effort to improve its product's quality. For this reason, many organizations are aiming to deploy or improve their testing process. To facilitate the software process improvement, many models have been developed in recent years, like IDEAL [McFeeley, 1996], ISO/IEC 15504 [ISO 15504,2005] and Pro2Pi [Salviano, 2006]. Also, there are specific reference models for defining requirements to testing processes, such as TMM [Burnstein, 2003], TIM [Ericson, 1996] e TPI [Koomen, 1999]. However, these models do not give guidelines for their deployment in a testing organization.

Although such process improvement models, as well as software test reference models, have been emerged, many Brazilian software companies still have a lot of difficulties to implement an effective software testing process improvement program. This is due to the fact that these models are “generic” and require many adaptations to treat the common difficulties found in software testing improvement programs. In this scenario, the approach proposed in this work will guide the companies to deploy the testing process in a simple and agile way, in order to reach better quality for their products.

Keywords: Software Testing, Process Improvement Model, Testing Maturity Model.

SUMÁRIO

<i>Introdução</i>	12
1.1 Motivação	12
1.2 Escopo e Contribuições Esperadas	15
1.3 Metodologia	16
1.4 Organização da dissertação	17
<i>Modelos para Melhoria de Processos</i>	18
2.1 Introdução	18
2.2 IDEAL	20
2.2.1 Visão Geral.....	20
2.2.2 Fases do IDEAL	21
2.3 ISO/IEC 15504-4	24
2.3.1 Visão Geral.....	24
2.3.2 Passos da ISO/IEC 15504-4	25
2.4 Pro2Pi	29
2.4.1 Visão Geral.....	29
2.4.2 Propriedades do Pro2Pi	30
2.4.3 Pro2Pi-Model	30
2.4.4 Pro2Pi-MEAS.....	31
2.4.5 Pro2Pi-Cycle	31
2.4.6 Pro2Pi-Work.....	32
2.5 Análise dos Modelos de Melhoria	34
2.6 Considerações Finais	36
<i>Metodologias Ágeis de Desenvolvimento de Software</i>	39
3.1 Introdução	39
3.2 Extreme Programming	40
3.2.1 Visão Geral.....	40
3.2.2 Valores de XP.....	41
3.2.3 Princípios Básicos	43
3.2.4 Práticas de XP	43
3.3 SCRUM	45
3.3.1 O Processo.....	46
3.3.2 Regras do SCRUM.....	47
3.4 Considerações Finais	50
<i>Modelo de Maturidade em Testes</i>	52

4.1 Introdução	52
4.2 Visão Geral do TMM.....	53
4.3 Níveis de Maturidade	54
4.3.1 Nível 1: Inicial.....	54
4.3.2 Nível 2: Definição	54
4.3.3 Nível 3: Integração	56
4.3.4 Nível 4: Gerenciamento e Medição	57
4.3.5 Nível 5: Otimização.....	58
4.4 Modelo de Avaliação.....	60
4.4.1 Seleção do Time de Avaliação e Treinamento	60
4.4.2 O Método de Avaliação.....	61
4.4.3 Questionário de Avaliação do TMM	63
4.5 Algoritmo de Classificação.....	64
4.6 Limitações e Dificuldades em utilizar o TMM.....	65
4.7 Considerações Finais	66
<i>Abordagem AITS.....</i>	<i>67</i>
5.1 Introdução	67
5.2 Visão Geral	68
5.3 Solução e Benefícios	71
5.3.1 Visão Geral das Fases da Abordagem	71
Iniciação	72
Diagnóstico.....	73
Identificar e Priorizar Ações de Melhoria.....	73
Implementar Ações de Melhoria	74
Verificar Resultados e Aprender	74
Institucionalizar a Melhoria	74
5.3.2 Elementos da Abordagem de Implantação	75
5.4 Detalhamento das Fases da AITS.....	76
5.4.1 Fase 1 – Inicialização	76
Objetivo	76
Atividades.....	77
Templates	78
Papéis	78
5.4.2 Fase 2 – Diagnóstico	79
Objetivo	79
Atividades.....	80
Templates	83
Papéis	84
5.4.3 Fase 3 – Identificar e Priorizar Ações de Melhoria	85
Objetivo	85
Atividades.....	85
Templates	89
Papéis	89
5.4.4 Fase 4 – Implementar Ações de Melhoria	90
Objetivo	90
Atividades.....	90
Templates	91

Papéis	91
5.4.5 Fase 5 – Verificar Resultados e Aprender	92
Objetivo	92
Atividades	92
Templates	93
Papéis	93
5.4.6 Fase 6 – Institucionalizar a Melhoria	94
Objetivo	94
Atividade	94
Templates	95
Papéis	95
5.4.7 Fase 7 – Acompanhamento do Programa de Melhoria	96
Objetivo	96
Atividades	96
Templates	97
Papéis	98
5.5 Considerações Finais	98
<i>Aplicação da Abordagem AITS</i>	<i>102</i>
6.1 Visão Geral	102
6.2 Estudo de Caso 1	103
6.3 Estudo de Caso 2	116
6.4 Análise das Práticas Ágeis nos Estudos de Caso	128
6.5 Considerações Finais	130
<i>Conclusão</i>	<i>132</i>
7.1 Considerações finais e Principais Contribuições	132
7.2 Dificuldades Encontradas	135
7.3 Proposta para trabalhos futuros	136
<i>Referências Bibliográficas</i>	<i>137</i>
<i>Anexo I – Documentos dos Estudos de Caso e Auxiliares</i>	<i>144</i>
<i>Anexo II – Templates</i>	<i>156</i>
<i>Anexo III – Questionário</i>	<i>167</i>

LISTA DE FIGURAS

<i>Figura 1 - Fases do IDEAL</i>	<i>20</i>
<i>Figura 2 - Passos para determinação da capacidade de processo [ISO 15504, 2005]</i>	<i>26</i>
<i>Figura 3 - Ciclo de melhoria da 15504-4.....</i>	<i>28</i>
<i>Figura 4 - Pro2Pi-Cycle.....</i>	<i>32</i>
<i>Figura 5 – Ciclo de melhoria de processo.....</i>	<i>36</i>
<i>Figura 6 - Processo do Scrum</i>	<i>47</i>
<i>Figura 7 - Áreas de Processo do TMM.....</i>	<i>59</i>
<i>Figura 8 - Modelo de Avaliação do TMM</i>	<i>63</i>
<i>Figura 9 - Fases da abordagem AITS.....</i>	<i>69</i>
<i>Figura 10 - Estrutura da Abordagem de Implantação</i>	<i>75</i>
<i>Figura 11 - Atividades da fase de Inicialização</i>	<i>79</i>
<i>Figura 12 - Atividades da Fase de Diagnóstico</i>	<i>85</i>
<i>Figura 13 - Técnica SWOT para pré-priorização de ações [Weber, 2005]</i>	<i>86</i>
<i>Figura 14 - Priorização de Ações [Weber, 2005]</i>	<i>87</i>
<i>Figura 15 - Plano de Ação</i>	<i>89</i>
<i>Figura 16 - Atividades da Fase de Planejar e Priorizar Ações.....</i>	<i>90</i>
<i>Figura 17 - Atividades da Fase de Implementar Ações de Melhoria</i>	<i>91</i>
<i>Figura 18 - Processo da Fase Verificar Resultados e Aprender</i>	<i>94</i>
<i>Figura 19 - Processo da Fase Institucionalizar Ações.....</i>	<i>95</i>
<i>Figura 20 - Processo da Fase de Acompanhar Programa de Melhoria.....</i>	<i>98</i>

LISTA DE TABELAS

<i>Tabela 1 - Análise do modelo IDEAL em relação aos critérios</i>	23
<i>Tabela 2 - Análise da ISO/IEC 15504-4 em relação aos critérios</i>	28
<i>Tabela 3 - Análise do Pro2Pi-Cycle em relação aos critérios</i>	33
<i>Tabela 4- Nível de satisfação para as sub-práticas</i>	64
<i>Tabela 5 - Classificação de satisfação dos objetivos de maturidade</i>	65
<i>Tabela 6 - Checklist de Avaliação</i>	81
<i>Tabela 7 - Mapeamento das atividades da AITS com conceitos e princípios ágeis</i>	99
<i>Tabela 8 - Avaliação do Processo de Teste do Estudo de Caso 1</i>	105
<i>Tabela 9 - Gráfico do Diagnóstico em relação as áreas de processo do Estudo de Caso 1</i>	105
<i>Tabela 10 - Plano de Ação</i>	108
<i>Tabela 11 - Métricas definidas para acompanhamento do programa</i>	113
<i>Tabela 12 - Duração das fases do programa</i>	113
<i>Tabela 13 - Esforço gasto em cada fase do programa</i>	114
<i>Tabela 14 - Resultado da avaliação após a implementação da AITS – Estudo de Caso 1</i>	115
<i>Tabela 15 - Áreas de processo após a implementação da AITS – Estudo de Caso 1</i>	115
<i>Tabela 16 - Avaliação do Processo de Teste do Estudo de Caso 2</i>	118
<i>Tabela 17 - Gráfico do Diagnóstico em relação as áreas de processo do Estudo de Caso 2</i>	118
<i>Tabela 18 - Ações Priorizadas</i>	120
<i>Tabela 19 - Plano de Ação Estudo de Caso 2</i>	121
<i>Tabela 20- Duração de cada fase do programa de melhoria</i>	125
<i>Tabela 21 - Resultado da avaliação após a implementação da AITS – Estudo de Caso 2</i>	126
<i>Tabela 22 - Áreas de processo após a implementação da AITS – Estudo de Caso 2</i>	126
<i>Tabela 23 - Número de bugs antes e durante a AITS</i>	127
<i>Tabela 24 - Mapeamento entre as Áreas de processo do TMM e as questões do checklist do AITS</i>	144
<i>Tabela 25 - Avaliação do Processo de Teste do Estudo de Caso 1</i>	148
<i>Tabela 26 - Avaliação do Processo de Teste do Estudo de Caso 2</i>	152

1

Introdução

Este capítulo apresenta uma visão geral desta dissertação. A Seção 1.1 apresenta a motivação deste trabalho. A Seção 1.2 demarca o escopo e as contribuições esperadas deste trabalho. A seção 1.3 contém uma descrição da metodologia utilizada no trabalho. Por fim, a Seção 1.4 fornece uma visão dos capítulos da dissertação.

1.1 Motivação

Nas últimas décadas, indústrias de software têm investido um esforço substancial na melhoria da qualidade de seus produtos. Este tem sido um trabalho difícil, já que o tamanho e a complexidade do software aumentam rapidamente, os usuários e clientes estão se tornando mais exigentes e o mercado cada vez mais competitivo [Vasconcelos et al, 2006].

Diante deste cenário, a indústria mundial de tecnologia da informação tem sofrido profundas transformações e investido, cada vez mais, em atividades de testes no intuito de aumentar a qualidade do software produzido. Pensando nisso, muitas empresas estão buscando implantar ou melhorar seus processos de teste. A melhoria de processo de testes é uma instanciação da melhoria de processos. Por isso, assumimos que as limitações e benefícios da melhoria de processo também se aplicam para o contexto de melhoria de processo de teste.

Pesquisas indicam que empresas que investiram em melhoria de processo de software obtiveram retorno de investimento da ordem de 5:1 e, em alguns casos, até 9:1 [Ste_06, 2008]. Diversas experiências como [Setúbal, 2007], [Ferreira, 2007] e

[Herbsleb, 1994], fornecem evidências de que a melhoria do processo traz diversos benefícios, tais como:

- Profissionais mais preparados;
- Software de maior qualidade;
- Habilidade de cumprir metas de prazos e custos;
- Habilidade de cumprir objetivos;
- Aumento da satisfação do cliente.

Apesar de todos estes benefícios, realizar um programa de melhoria de processo não é uma atividade trivial, exige conhecimentos, habilidades e métodos específicos.

Com o intuito de facilitar a melhoria de processo de software, diversos modelos têm surgido nas últimas décadas. São modelos que fornecem diretrizes para guiar iniciativas de programas de melhoria de software, tais como IDEAL [McFeeley, 1996], ISO/IEC 15504 [ISO 15504, 2005] e Pro2Pi [Salviano, 2006]. Existem também modelos de referência específicos para a definição de requisitos necessários a um Processo de Testes, tais como: TMM [Burnstein, 2003], TIM [Ericson, 1996] e TPI [Koomen, 1999]. No entanto, tais modelos não fornecem diretrizes para a sua implantação em uma organização de testes.

Embora tenham surgido tais modelos de melhoria de processo, bem como os modelos de referência para processos de testes, e haja resultados encorajadores, grande parte da indústria de software ainda apresenta muitas dificuldades em realizar um programa de melhoria de processo de testes de software efetivo. Isto ocorre por uma série de motivos, conforme descrito em [Niazi, 2003], [MCT_01, 2001], [Santana, 2007] e [Carosia, 2003]. Os principais motivos são os seguintes:

- Resistência a mudanças;
- Falta de recursos financeiros;
- Conflito de prioridades entre projetos;
- Dificuldade em medir sucesso;
- Benefícios geralmente obtidos a médio e longo prazo;
- Muitas mudanças culturais e organizacionais em pouco tempo;

- Falta do uso de padrões de qualidade e falta de conhecimento nos modelos utilizados.

Dentre estes motivos, há o fato destes modelos serem genéricos e exigirem diversas adaptações para tratar as dificuldades comumente encontradas em um programa de melhoria de testes. Adaptar, para a necessidade da organização, um modelo detalhado não é uma tarefa fácil, pois é necessário um conhecimento do modelo completo para que se possa escolher quais atividades serão realmente úteis. Devido ao grande escopo e a ao alto grau de detalhamento destes modelos de melhoria, esta é uma tarefa bastante trabalhosa e complexa. Brodman [Brodman, 1994] afirma que um número alto de programas de melhoria de software baseado em modelos como CMMI [CMMI, 2001] e IDEAL [McFeeley, 1996] falham e que 53% dos casos de falha estão relacionados à complexidade e tamanho destes modelos. Além disso, modelos de referência, como o TMM, não fornecem o “como fazer”. Fornecem somente “o que fazer” para que a empresa possa atingir um dos seus níveis de maturidade.

Diante deste cenário, é necessária a elaboração de uma abordagem simples e ágil que combine os benefícios dos modelos de melhoria de processo com os benefícios fornecidos pelos modelos de referência em testes, no nosso caso o TMM, que foi escolhido por ser o modelo focado em testes mais conhecido [Freesz, 2004], [Reffson, 2006], [Santos, 2006]. Tal abordagem deve atender às necessidades das empresas, que necessitam melhorar seus processos de testes rapidamente mesmo, possuindo todas as limitações apresentadas acima, para conseguir competir no mercado do qual elas fazem parte.

A abordagem proposta irá guiar as empresas de maneira simples e ágil a implantar e melhorar seus processos de testes de software, possibilitando assim o alcance de níveis melhores de qualidade de seus produtos.

1.2 Escopo e Contribuições Esperadas

Este trabalho tem como objetivo desenvolver uma Abordagem de Implantação para Teste de Software baseada em Metodologias Ágeis (AITS), contribuindo com a área de implantação e melhoria de processos de teste de software. Tal abordagem foi elaborada a partir da combinação dos modelos de melhoria de processo existentes, de forma que seja simples, dinâmica e trate as principais limitações encontradas em programas de melhoria de software [Niazi, 2003], [MCT_01, 2001], [Santana, 2007] e [Carosia, 2003]. Para atingir este objetivo, o modelo proposto tem como base práticas de metodologias ágeis usufruindo dos benefícios que estas metodologias podem trazer [Begel, 2007], [Teles, 2005], assim como o modelo de referência de processo de teste, TMM. Nesse sentido, as hipóteses a seguir são tomadas como base para o desenvolvimento desse trabalho:

1. É possível estabelecer processo de testes em empresas de software, considerando suas limitações, como, por exemplo: falta de recursos humanos e financeiros;
2. O estabelecimento de processo de testes permite a inserção de ações na organização que podem melhorar a qualidade de seu produto;
3. O estabelecimento de processo de testes de software favorece a criação de uma cultura voltada para processos e melhoria contínua;

Espera-se que os resultados alcançados na execução do processo de teste sejam mais previsíveis e a organização alcance uma série de benefícios, como, por exemplo, obtenção de resultados em um curto espaço de tempo, melhor gerenciamento dos riscos, *feedback* freqüente e contínuo, possibilidade de priorização, definição clara dos papéis e das responsabilidades que cada um desempenha no processo, melhoria no entendimento do processo, entre outros.

De forma explícita, os itens seguintes são apresentados como resultado do desenvolvimento desse trabalho:

1. Levantamento do estado da arte na área de melhoria de processo de teste de software;
2. A descrição detalhada de uma abordagem para estabelecer processo de teste de software;
3. Execução de dois estudos de caso estabelecendo processo de teste de software, incluindo os resultados e as experiências obtidas.

1.3 Metodologia

O processo deste trabalho foi guiado pela metodologia baseada na abordagem Indústria-como-laboratório (*Industry as laboratory*) [Potts, 1993], na qual pesquisadores identificam problemas através do envolvimento próximo com projetos da indústria, criam e avaliam soluções em uma atividade quase indivisível de pesquisa.

Para trabalhos em Engenharia de Software, Potts defende a abordagem “indústria-como-laboratório” (*industry-as-laboratory*) como uma abordagem complementar à tradicional “pesquisa-depois-transfere” (*research-then-transfer*). Na primeira abordagem, as pesquisas são iniciadas a partir de um problema prático, e mais tarde serão refinadas com a sua aplicação em estudos de casos contínuos e incrementais.

Segundo Potts, “os domínios de pesquisa e aplicação tendem a convergir, e a transferência de tecnologia e o processo de avaliação são realizados nos estágios iniciais do programa de pesquisa” [Potts 1993, p. 22].

Baseado nisto, a metodologia deste trabalho foi realizada em três fases: levantamento, desenvolvimento e formalização. A primeira fase foi realizada de maio de 2006 a setembro de 2006 e focou no levantamento e estudo do estado da arte e dos problemas enfrentados pelas empresas de software na implantação e melhoria de processos de testes de software.

A segunda fase, denominada desenvolvimento, foi realizada de julho de 2006 a dezembro de 2007, e teve por objetivo definir e aplicar a abordagem para implantação e melhoria de processos de testes em empresas de software, com base nos modelos para programa de melhoria IDEAL [McFeeley, 1996], ISO/IEC 15504 [ISO 15504, 2005] e

Pro2Pi [Salviano, 2006] e no modelo de maturidade de testes TMM [Burnstein, 2003]. A primeira versão da abordagem definida foi aplicada em uma pequena empresa do Recife, o que resultou em uma experiência com inúmeros resultados positivos [Diniz et al, 2007]. Com o *feedback* coletado através do primeiro estudo de caso, foi possível refinar e detalhar a abordagem gerando uma versão intermediária [Ochner, 2007]. Com esta versão foi realizado outro estudo de caso em uma média empresa também de Recife, gerando a última versão da abordagem proposta neste trabalho.

A terceira e última fase, denominada formalização, foi realizada entre setembro de 2007 e janeiro de 2008. Nesta fase, a abordagem foi formalizada como um exemplo de abordagem para implantação e melhoria de processo de testes de software, composta por alguns elementos essenciais: fases, atividades, templates e papéis.

1.4 Organização da dissertação

Esta dissertação está organizada em sete capítulos e três apêndices, dos quais esse é o primeiro capítulo. No Capítulo 2 é apresentado o estado da arte da área de melhoria de processo de software, incluindo modelos e abordagens para melhoria de processo de software. No capítulo 3 são apresentadas as metodologias ágeis que serão utilizadas neste trabalho. No capítulo 4, o modelo de maturidade de teste TMM é apresentado detalhadamente. No Capítulo 5 é descrita, em detalhes, a abordagem de implantação de testes proposta. No capítulo 6, são descritas as experiências de utilização da versão intermediária e da versão final da abordagem e os resultados obtidos através de suas aplicações. No Capítulo 7 é apresentada a conclusão do trabalho com as considerações finais, bem como um resumo das contribuições e proposta para trabalhos de pesquisa futuros. Nos apêndices são descritos todos os templates propostos nas atividades definidas na abordagem AITS, as tabelas auxiliares utilizadas ao longo do trabalho e o questionário de avaliação de satisfação dos participantes da implantação da abordagem AITS.

2

Modelos para Melhoria de Processos

Este capítulo apresenta alguns conceitos básicos dos principais modelos atuais de melhoria de processo. Na Seção 2.1, é apresentada uma visão geral de melhoria de processos e sua importância. Na Seção 2.2, o IDEAL é apresentado em detalhes. Na seção 2.3 e 2.4, a ISO/IEC 15504-4 e o Pro2Pi são apresentados respectivamente. Na seção 2.5, uma análise dos modelos escolhidos é realizada. E por último, na seção 2.6, são expostas as considerações finais do capítulo.

2.1 Introdução

Melhoria de Processo de Software é uma abordagem sistemática para melhorar processos de software a fim que as empresas alcancem as metas de negócios mais efetivamente (ISO, 2003). A melhoria de processo tem mostrado ser uma abordagem viável e eficiente trazendo inúmeros benefícios para as organizações. No entanto, iniciar um programa de melhoria de software não é tarefa trivial e exige conhecimentos, habilidades e métodos específicos.

Nos últimos anos, muitos modelos foram elaborados com o objetivo de guiar as organizações no caminho da melhoria de processo de software provendo um roteiro racional para isto. Alguns dos principais modelos serão apresentados neste capítulo.

Uma comparação entre os modelos será realizada baseada em critérios que endereçam as limitações encontradas em um programa de melhoria:

- Custos de acordo com a realidade da empresa: Uma das limitações apresentadas pelas empresas que fazem parte do contexto deste trabalho é a falta de recursos financeiros.
- Ser de fácil compreensão: É importante que o modelo seja detalhado passo a passo e que sua aplicação seja de fácil compreensão, provendo descrições detalhadas das fases, atividades e templates. Assim o tempo de entedimento do modelo é reduzido e os envolvidos ficam mais motivados com a compreensão do modelo;
- Ser flexível: É importante que o modelo seja flexível o suficiente para se adaptar ao contexto da empresas mantendo a cultura da organizacional e evitando resistência dos envolvidos e o choque cultural;
- Baseado em modelos reconhecidos: É importante que o modelo seja baseado em modelos reconhecidos internacionalmente, usufruindo assim das melhores práticas e estando de acordo com os principais modelos reconhecidos pelo mercado;
- Resultados rápidos: É importante que o modelo gere resultados em um curto espaço de tempo para que os envolvidos continuem motivados com o programa;
- Modelo incremental: É importante que o modelo considere que o programa de melhoria seja realizado de forma incremental. Esta abordagem ajuda a introduzir as mudanças aos poucos e de forma gradativa. Sendo assim, a resistência a mudanças e o choque cultural são minimizados.

2.2 IDEAL

2.2.1 Visão Geral

O IDEAL é um modelo para programa de melhoria de processo de software que foi desenvolvido pelo SEI (Software Engineering Institute) [IDEAL, 2007]. O objetivo deste modelo é fornecer um guia para direcionar iniciativas de programas de melhoria através de um longo e integrado plano para iniciação e gerenciamento. Este modelo define cinco fases: Iniciação, Diagnóstico, Estabelecimento, Ação e Lições Aprendidas. Cada fase é constituída em média por 10 tarefas e algumas possuem 5 ou 6 sub-tarefas. As principais atividades de cada fase são exibidas na figura 1.

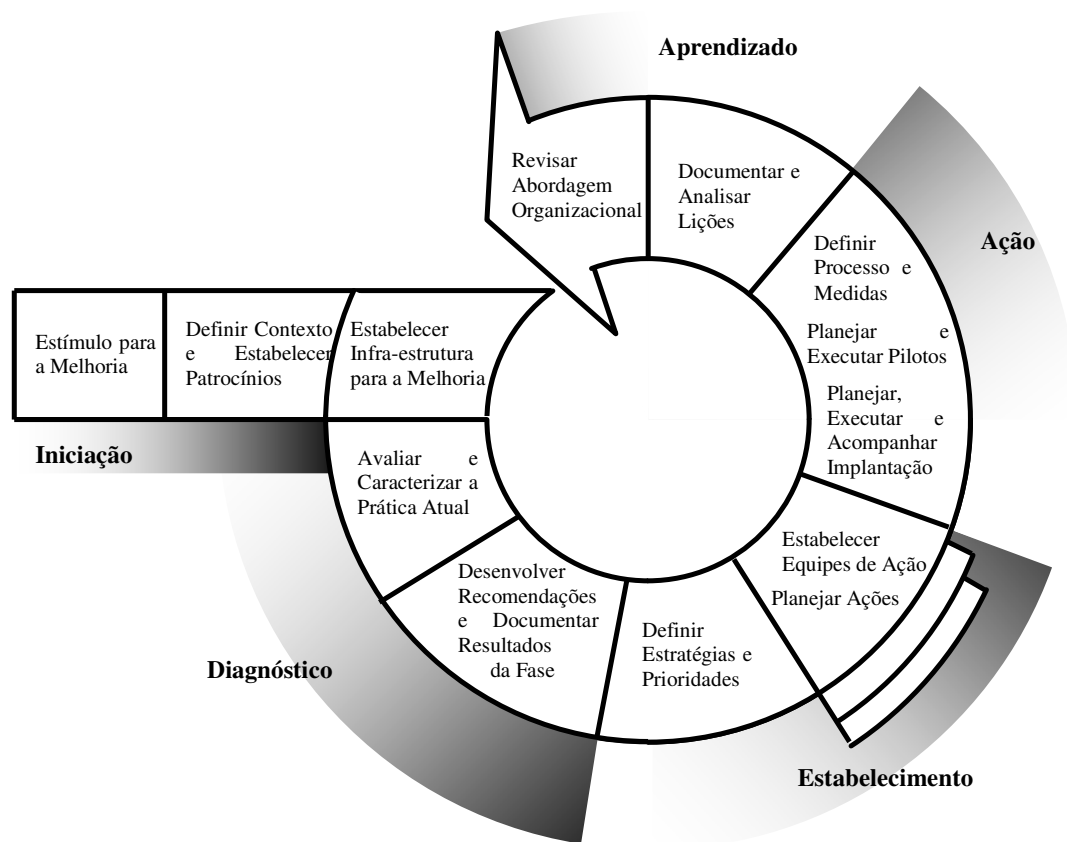


Figura 1 - Fases do IDEAL

2.2.2 Fases do IDEAL

Iniciação

A fase de iniciação do modelo IDEAL é o ponto de partida para o programa de melhoria. É quando a infra-estrutura da melhoria é estabelecida, papéis, responsabilidades e recursos são inicialmente definidos. Nesta fase, o plano do programa de melhoria é criado para guiar a organização na finalização das fases de Iniciação, Diagnóstico e Estabelecimento. A aprovação para a iniciativa de melhoria é obtida através de um comprometimento de recursos para o trabalho futuro. Além disso, as metas gerais para o programa de melhoria são definidas durante esta fase. Elas são identificadas através das necessidades de negócios da organização, refinadas e realizadas durante a fase de Estabelecimento do IDEAL.

A infra-estrutura inicial para suportar e facilitar o programa será estabelecida também durante esta fase. Dois grupos são tipicamente estabelecidos, o grupo de Gerenciamento (*MSG - Management Steering Group*) e o grupo de processos de engenharia de software (*SEPG – Software Engineering Process Group*). Também durante a fase de Iniciação, planos são realizados para comunicar a iniciativa do programa de melhoria, e é sugerido que avaliações organizacionais sejam realizadas para determinar a disposição da organização em começar um programa de melhoria.

Diagnóstico

A fase de diagnóstico do modelo IDEAL inicia a organização no caminho da melhoria contínua do processo de software. Esta fase prepara o trabalho para a próxima fase. As atividades de avaliação são desempenhadas para estabelecer um *baseline* do estado atual do processo da organização. Os resultados e recomendações dos avaliadores são consolidados com outras iniciativas de melhoria para inclusão no plano de ação.

Estabelecimento

Durante a fase de Estabelecimento, as questões que a organização decidiu implementar são priorizadas; estratégias para as soluções também são desenvolvidas. A versão inicial

do plano de melhoria será elaborada de acordo com a visão da organização, plano estratégico de negócios e lições aprendidas de experiências passadas.

Durante esta fase, objetivos mensuráveis são definidos a partir de objetivos gerais que foram definidos na fase de Iniciação; estes objetivos mensuráveis serão incluídos na versão final do plano de ação do programa de melhoria.

Métricas necessárias para monitorar o processo são definidas, os recursos são obtidos e treinamentos fornecidos para o grupo de trabalho técnico (*TWG – Technical Work Group*). O plano de ação definido irá guiar as atividades de melhoria endereçando a prioridade das recomendações da fase de Diagnóstico. Também durante esta fase, templates de plano de ação são criados e disponibilizados para os grupos de trabalhos técnicos (TWGs) utilizá-los.

Ação

Na fase de Ação do modelo IDEAL, soluções endereçadas para os pontos de melhoria diagnosticados durante a fase de Diagnóstico são criadas, testadas e implantadas na organização. Planos serão desenvolvidos para execução de pilotos a fim de testar e avaliar os processos novos ou alterados.

Depois que os pilotos são realizados com sucesso e determinada a aceitação de adoção quanto ao novo/alterado processo, a implantação, institucionalização e planos para extensão são desenvolvidos e executados.

Aprendizagem

O objetivo da fase de aprendizagem é fazer com que o próximo passo através do modelo IDEAL seja mais efetivo. Neste momento, soluções já foram desenvolvidas, lições já foram aprendidas, métricas coletadas e objetivos alcançados. Os artefatos são adicionados à base de dados do processo que irá se tornar uma fonte de informação para as pessoas envolvidas no próximo passo do modelo.

A tabela 1 exibe a análise do modelo IDEAL em relação aos critérios estabelecidos na seção 2.1.

Tabela 1 - Análise do modelo IDEAL em relação aos critérios

Crítérios	Avaliação	Observações
Custos de acordo com a realidade da empresa		Custos de implementação bastante elevados, necessitando a criação de diferentes grupos para a realização da melhoria e de membros <i>full time</i> .
Ser de fácil compreensão		Modelo bem detalhado, especifica fases, atividades e templates, mas possui muitas atividades e sub-atividades que exigem muito tempo e atenção para o seu entendimento, torna o processo burocrático demais e dificulta sua aplicação.
Ser flexível		Modelo genérico que pode ser customizado para diferentes organizações, embora os custos das adaptações sejam bastante elevados [Rico, 2002], [Anacleto, 2004], [Santana, 2007]. Pode ser utilizado com diferentes modelos, tais como CMMI, TMM, MPS.BR.
Baseado em modelos reconhecidos		É baseado no modelo PDCA [Deming, 1986]
Resultados Rápidos		Resultados obtidos a longo prazo, geralmente de 3-5 anos [McFeeley, 1996].
Modelo Incremental		É aplicado de forma incremental, embora um ciclo do modelo IDEAL possa durar muito tempo.

Legenda:



Atende completamente



Atende parcialmente



Não atende



Informação desconhecida

2.3 ISO/IEC 15504-4

A ISO/IEC 15504-4 é um guia para orientação de melhoria de processo que está definido na ISO/IEC 15504. A ISO/IEC 15504 fornece um framework para avaliação de processo que pode ser utilizado também como modelo de referência para melhoria de processo de software. Este framework é composto basicamente por uma estrutura e requisitos para avaliação de processos possuindo 5 documentos, onde a parte 1 descreve os conceitos e vocabulário utilizado, as partes 2 e 3 descrevem o framework e orientações para o uso do framework, e as partes 4 e 5 descrevem orientações para utilização de modelos para melhoria ou capacidade e o modelo exemplo. A ISO/IEC 15504 criou a chamada arquitetura contínua de modelos de referência de processo, onde existem duas dimensões: a primeira na qual são definidos processos e a segunda na qual são definidos níveis de capacidade.

Neste capítulo será apresentada mais detalhadamente a ISO/IEC 15504-4 [ISO/IEC 15504-4, 2004] onde é descrito um guia para orientação da melhoria de processo. Somente a parte 4 da ISO/IEC 15504 foi escolhida para detalhamento neste capítulo, pois é a que apresenta grande relevância para a abordagem proposta neste trabalho.

2.3.1 Visão Geral

A ISO/IEC 15504-4 é um guia para orientação de melhoria de processo, o ciclo de melhoria da sua versão atual é composto por oito passos: examinar metas de negócio, iniciar ciclo de melhoria, avaliar capacidade corrente, desenvolver plano de ação, implementar melhorias, confirmar melhorias, manter melhorias e monitorar desempenho.

2.3.2 Passos da ISO/IEC 15504-4

Examinar Necessidades da Organização

Neste passo é necessário que os objetivos de negócio da organização e os estímulos para melhoria sejam determinados. Determinar um objetivo de melhoria envolve primeiramente identificar qual modelo de referência de processo se encaixa melhor nos objetivos da organização. Isso inclui definir um perfil alvo de processo, que apresentará processos que serão avaliados e guiará a identificação de ações de melhorias que serão mais efetivas.

É necessário também obter o comprometimento gerencial e financeiro com o programa de melhoria. Ele deve fazer parte do plano estratégico da organização e o orçamento para o programa deve ser definido.

Iniciar Processo de Melhoria

Para o programa de melhoria deve ser definido patrocínio, gerenciamento de projeto, orçamento, *milestones* e responsáveis. O programa deve ser gerenciado de acordo com o processo de gerenciamento alinhado ao modelo de avaliação que é utilizado. Um plano para o programa de melhoria deve ser elaborado contendo as seguintes informações: história e o estado atual das atividades do processo de melhoria, objetivos do programa, escopo organizacional – limites organizacionais do programa de melhoria, escopo do processo – os processos que serão melhorados, ciclo de vida do processo, papéis e responsabilidades dos envolvidos, recursos, milestones, riscos associados com o programa e atividades que serão desenvolvidas.

Avaliar Processo

A avaliação do processo inicia seguindo o guia de avaliação de processo definido na ISO/IEC 15504-3 [ISO 15504, 2005]. A figura 2 define os passos para determinar a capacidade de um processo utilizando o processo de avaliação descrito na ISO/IEC 15504-2 e ISO15504-3 [ISO 15504, 2005].

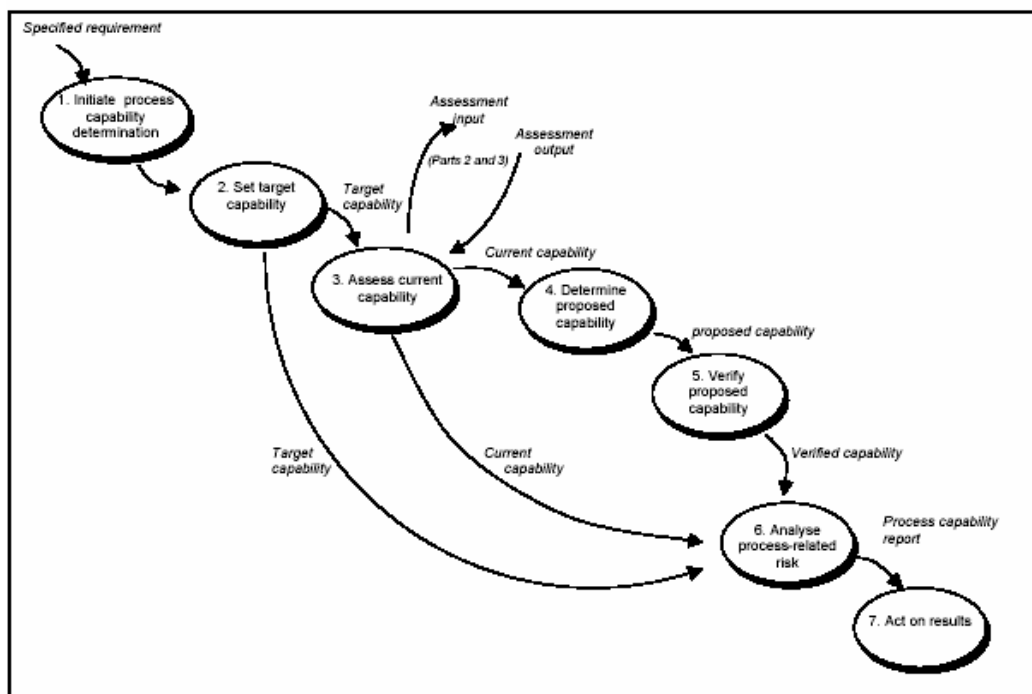


Figura 2 - Passos para determinação da capacidade de processo [ISO 15504, 2005]

Desenvolver Plano de Ação

O resultado da avaliação é analisado através dos objetivos estratégicos da organização. Neste passo, é necessário realizar as seguintes ações: identificar, analisar e listar as áreas de melhorias, definir objetivos de melhoria específicos e elaborar um plano de ação.

Implementar Melhorias

O plano de Ação é implementado a fim de melhorar os processos da organização. A implementação pode ser simples ou complexa dependendo do conteúdo do plano de ação e das características da organização. Quatro tarefas principais estão associadas com esta fase: selecionar uma estratégia para implementação, preparar um Plano de Implementação detalhado, colocar o plano de implementação em prática e monitorar o progresso através do plano.

Confirmar Melhorias

Depois que a implementação das melhorias for finalizada, a organização deve: confirmar que os objetivos planejados foram alcançados e os benefícios esperados foram entregues, confirmar que os processos e práticas apropriados foram adotados, requisitar uma avaliação de processo para confirmar que a capacidade de processo desejada foi estabelecida.

Manter Melhorias

Depois que as melhorias foram confirmadas, os processos precisam ser mantidos no novo nível de capacidade. Os processos melhorados devem ser usados por todos os membros da organização. Isto requer um gerenciamento para monitorar os processos melhorados. Responsabilidades pelo monitoramento devem ser definidas. O monitoramento deve ser feito utilizando medições apropriadas.

Monitorar Desempenho

O desempenho dos processos da organização deve ser continuamente monitorado e novas melhorias nos processos devem ser iniciadas como parte de um programa contínuo de melhoria. O programa de melhoria deve ser regularmente revisado pela área gerencial para assegurar que: o mesmo, incluindo seus objetivos, está de acordo com os objetivos de negócios da organização; os processos são melhorados com base na experiência; e a melhoria contínua se torne parte dos valores, atitudes e comportamento da organização.

A figura 3 exibe o ciclo de melhoria proposto.

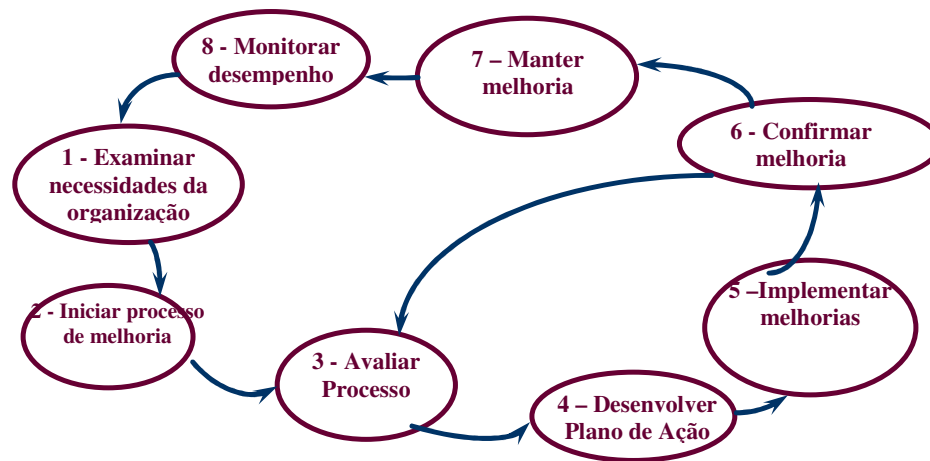


Figura 3 - Ciclo de melhoria da 15504-4

Na tabela 2 é realizada a análise da ISSO/IEC 15504-4 em relação aos critérios estabelecidos na seção 2.1.

Tabela 2 - Análise da ISO/IEC 15504-4 em relação aos critérios

Crítérios	Avaliação	Observações
Custos de acordo com a realidade da empresa	✗	Não foram encontradas informações específicas sobre o custo de aplicação do modelo. No entanto, diversos trabalhos foram encontrados necessitando da utilização de consultorias externas para a aplicação do modelo. Isso implica em aumento do custo de aplicação.
Ser de fácil compreensão	■	Modelo de Melhoria não especifica atividades e <i>templates</i> . Modelo de avaliação complexo e não apresenta um método de avaliação.
Ser flexível	●	Modelo genérico que pode ser customizado para diferentes organizações, embora as adaptações possam ser bastante custosas [Rico, 2002], [Anacleto, 2004], [Santana, 2007]. Pode ser utilizado com a ISO/IEC 12207.
Baseado em modelos	✗	Não foram encontradas informações específicas.

reconhecidos		
Resultados rápidos		Não foram encontradas informações específicas sobre o tempo de obtenção dos resultados. No entanto, foram encontrados alguns trabalhos que são adaptações da ISO/IEC 15504-4 e apontam a obtenção de resultados em um curto espaço de tempo como uma de suas vantagens em relação à ISO/IEC 15504-4.
Modelo incremental		É aplicado de forma incremental.

2.4 Pro2Pi

2.4.1 Visão Geral

Pro2Pi é uma abordagem para melhoria de processos de software dirigida por perfis de capacidade de processo. Para o Pro2Pi, um perfil de capacidade representa os requisitos e orientações para a melhoria do processo de uma unidade organizacional.

Uma melhoria de processo com o Pro2Pi busca alinhar um perfil de capacidade de processo com boas práticas selecionadas de modelos para melhoria de processos. Segundo Salviano [Salviano, 2006], a abordagem Pro2Pi pode ser descrita através de 4 componentes:

- Um perfil de capacidade de processo que deve possuir um conjunto de oito propriedades, sendo elas: relevante, oportuno, viável, sistêmico, representativo, rastreável a modelos relevantes, específico e dinâmico;
- Um modelo que permite a representação de uma abstração segundo o aspecto capacidade de processo do processo atual ou do processo alvo de uma organização;
- Um conjunto inicial de medições relacionadas à Pro2Pi, incluindo uma medição da complexidade e da viabilidade de um Pro2Pi e;

- Um ciclo de melhoria com PRO2PI.

2.4.2 Propriedades do Pro2Pi

Para o Pro2Pi, para que um Perfil de Capacidade de Processo seja útil para a melhoria de processo, ele deve ter as seguintes propriedades: relevante, oportuno, viável, sistêmico, representativo, rastreável a modelos relevantes, específico e dinâmico.

As cinco primeiras propriedades devem estar presentes em todos os perfis. As outras propriedades podem ser encontradas em apenas uma parte do perfil ou até mesmo não serem aplicadas a um determinado perfil [Salviano, 2006].

2.4.3 Pro2Pi-Model

O Pro2Pi possui ainda um modelo que é denominado de Pro2Pi-Model, este modelo deve atender aos seguintes requisitos:

- Reúne os elementos dos modelos de capacidade de processo mais relevantes, tais como CMMI [CMMI, 2001], MR-MPS [MPS, 2006] e ISO15504-5 [ISO15504, 2005], permitindo assim a representação de elementos de qualquer um destes modelos.
- Possui uma referência conceitual adequada na linha de Engenharia Dirigida a modelos [MDA, 2007].

Segundo Salviano [Salviano, 2006], um Pro2Pi é um modelo de capacidade de processo que foi baseado em dois frameworks para modelo de capacidade de processo: ISO15504 [ISO15504, 2005] e o CMMI [CMMI, 2001].

Um Pro2Pi-Model pode ser considerado como um metamodelo de um processo baseado na abstração de perfis de capacidade de processo.

2.4.4 Pro2Pi-MEAS

Pro2Pi-MEAS é um conjunto de medições relacionadas ao Pro2Pi. As medições definidas foram baseadas na norma ISO/IEC 15939 [ISO 15939, 2002], no PSM [Aguilar, 2002], nas áreas de processo de medição e análise do CMMI [CMMI, 2001] e ISO 15504 [ISO15504, 2005]. As medições definidas foram:

- Uma medida derivada da complexidade de uma melhoria de processo em relação à situação atual de uma unidade organizacional (DM-PIC), em termos de unidade de melhoria de processo;
- Uma medida derivada da efetividade de investimento para melhoria de processo em uma unidade organizacional (DM-IEPI), em termos de Unidades de Eficiência de Investimento para Melhoria de Processo;
- O produto de informação denominado Viabilidade de PRO2PI, em termos de uma interpretação de DM-PIC e DM-IEPI.

2.4.5 Pro2Pi-Cycle

O Pro2Pi-Cycle é um processo para um ciclo de melhoria com Pro2Pi. Ele foi baseado nos modelos IDEAL [McFeeley, 1996] e ISO/IEC 15504 [ISO15504, 2005]. Um ciclo de melhoria Pro2Pi consiste de seis fases sequenciais, sendo elas: iniciar ciclo de melhoria, avaliar práticas correntes, planejar ações de melhoria, realizar ações de melhoria, preparar institucionalização, institucionalizar melhoria, e de uma atividade para definição e utilização do Pro2Pi que pode ser utilizado em qualquer uma das fases definidas. Para o Pro2Pi-Cycle a referência principal para a melhoria é perfil de capacidade de processo. Os objetivos do Pro2Pi-Cycle são:

- Definir os objetivos da melhoria através da identificação e análise dos objetivos, estratégias e contexto da organização;
- Avaliar e melhorar um perfil de capacidade através de critérios de qualidade estabelecidos;

- Identificar modelos de capacidade de processo relevantes para a organização;
- Definir um perfil de capacidade para a organização;
- Utilizar o perfil definido para orientar o entendimento do processo e orientar as ações de melhoria;
- Atualizar o perfil sempre que houver desvios identificados.

A figura 4 exibe as fases e outros elementos estabelecidos pelo Pro2Pi-Cycle.

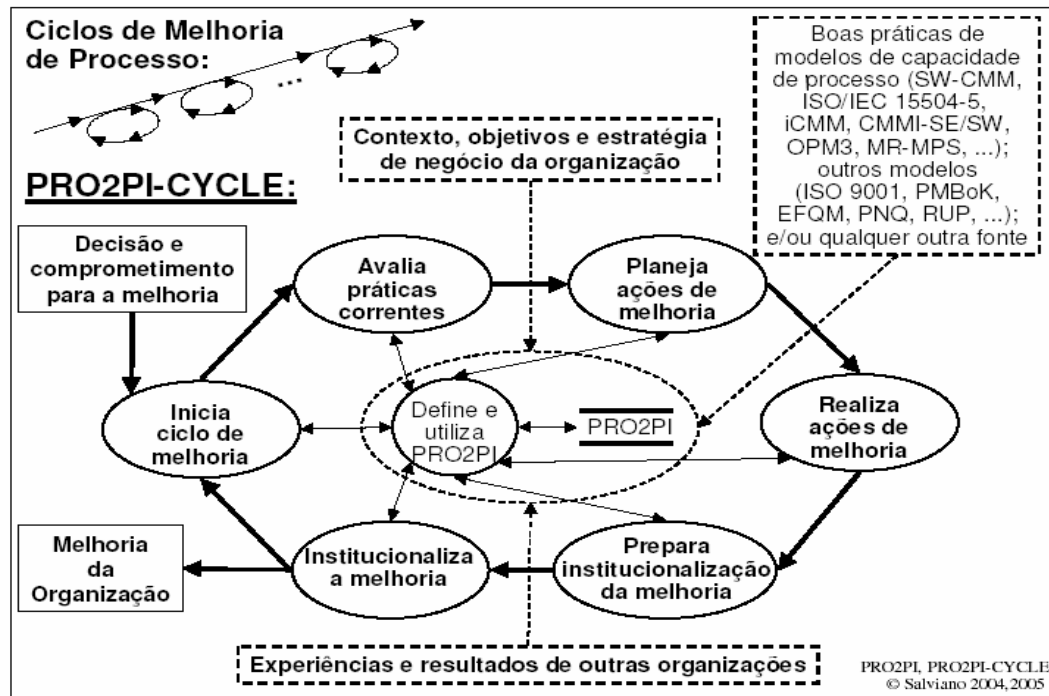


Figura 4 - Pro2Pi-Cycle

2.4.6 Pro2Pi-Work

Pro2Pi-Work é o método para estabelecimento do Pro2Pi utilizado nas atividades das três primeiras fases do Pro2Pi-Cycle. O objetivo principal do Pro2Pi-Work é estabelecer um perfil de capacidade de processo como guia para um ciclo de melhoria de processo. Ele é composto por quatro fases: preparação do trabalho, escolha do Pro2Pi, orientações do Pro2Pi e conclusão do trabalho.

A fase de preparação do trabalho consiste em conhecer a unidade organizacional onde será iniciado o ciclo de melhoria. Para isto, é necessário que sejam obtidas e analisadas informações sobre a unidade organizacional. Uma vez feito isso, é hora de preparar para as atividades da próxima fase.

Na segunda fase, é importante identificar fatores de negócio e organizacionais da unidade para que a relevância dos processos seja obtida e o Pro2Pi escolhido. Feito a escolha do Pro2Pi, é necessário então que ele seja apresentado.

Na fase de orientações para Pro2Pi, uma avaliação das práticas correntes é realizada e em seguida orientações para melhoria são elaboradas. As orientações são consolidadas e o Pro2Pi é revisado.

Na quarta fase, o trabalho é concluído com a elaboração e entrega de um relatório final e a avaliação do trabalho realizado.

Na tabela 3, é realizada a análise do modelo Pro2Pi em relação aos critérios estabelecidos na seção 2.1.

Tabela 3 - Análise do Pro2Pi-Cycle em relação aos critérios

Crítérios	Avaliação	Observações
Custos de acordo com a realidade da empresa.		Não foram encontradas informações específicas sobre o custo de aplicação do modelo. No entanto, diversos trabalhos encontrados [Salviano, 2006] afirmavam que os custos de melhoria seguindo o modelo Pro2pi-Cycle foram baixos.
Ser de fácil compreensão		Especifica fases e atividades, mas não define templates, dificultando assim sua aplicação.
Ser flexível		Modelo que pode ser aplicado para micro e pequenas empresas. Necessita de uma determinação de um perfil, o que para o contexto deste trabalho não faz sentido.

Baseado em modelos reconhecidos	●	Baseado no IDEAL e ISO/IEC 15504.
Resultados rápidos	●	Fornece a obtenção de resultados em um curto espaço de tempo, em média de 4 - 6 meses [Salviano, 2006].
Modelo incremental	●	É aplicado de forma incremental.

2.5 Análise dos Modelos de Melhoria

Para o contexto definido neste trabalho, os modelos apresentados possuem limitações que dificultam a sua aplicação na melhoria do processo de testes em empresas de desenvolvimento de software que possuem as restrições apresentadas na seção 1.1. Segundo uma análise realizada pela autora e conforme descrito em [McFeeley, 1996], [Rico, 2002], [Anacleto, 2004], [Santana, 2007] e [Carosia, 2003], as seguintes limitações foram encontradas:

IDEAL

- Dificuldade de se ter membros com dedicação full-time para o programa;
- Necessidade de criação de grupos diferentes para realização da melhoria de processo dentro da organização (ex: SEPG, TWG e MSG). Sendo assim, é necessário um número alto número de funcionários ou o acúmulo de papéis para a realização da melhoria;
- Modelo com muitas tarefas, sub-tarefas e artefatos definidos por fase, necessitando um esforço e conhecimentos substanciais para sua adaptação de acordo com a necessidade de cada empresa.
- Duração em média de 3 a 5 anos, dependendo do objetivo da organização. A cada novo ciclo é necessário realizar todas as fases novamente fazendo com que o ciclo de melhoria seja extenso e os resultados sejam percebidos a longo prazo;

- Custo bastante elevado.

ISO/IEC 15504-4

- Modelo que não especifica atividades e templates, dificultando assim sua aplicação;
- A avaliação definida é baseada no modelo de avaliação ISO/IEC 15504-2/3, que possui um processo para determinar a capacidade dos processos avaliados. Uma avaliação apoiada no modelo de avaliação ISO/IEC 15504-2/3 não é trivial e pode consumir muito tempo e recursos tornando-se muito custosa para a organização;
- A norma não apresenta, de maneira explícita, um método de avaliação, somente estabelece requisitos mínimos para que o processo de avaliação esteja em conformidade;
- Direcionado para modelos de capacidade de processo, dificultando a integração com modelos que possuem o conceito de maturidade, como é o caso do TMM;

PRO2PI

- Necessita da determinação de um de Perfil de Capacidade, porém para o contexto deste trabalho, a empresa já sabe previamente qual o processo a ser melhorado (Testes). Portanto, determinar um perfil de capacidade não faria sentido. Por isso, toda a fase relacionada com a determinação de perfis é irrelevante (Pro2pi-Work). Mesmo que cada área de processo definida pelo TMM pudesse fazer parte de um perfil, fazer esta adaptação não seria trivial, já que as áreas de processos do TMM são totalmente dependentes umas das outras, estando todas relacionadas a uma mesma disciplina.

A figura 5 apresenta os ciclos de melhoria dos modelos apresentados neste capítulo. .

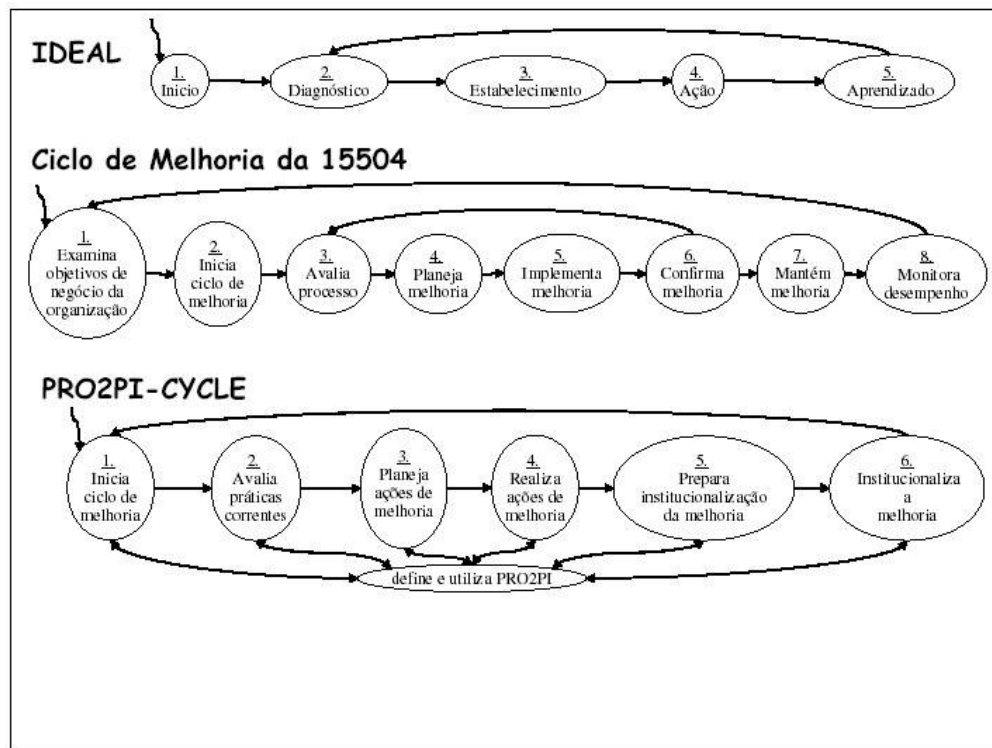


Figura 5 – Ciclo de melhoria de processo

2.6 Considerações Finais

O objetivo deste capítulo foi apresentar os principais modelos para melhoria de processo e analisar suas características no contexto de implantação de testes de software.

Como conclusão desta análise foi constatado que os modelos de melhoria de processos possuem diversas semelhanças entre si conforme exibido na figura 5. Embora apresentem vários benefícios [Setúbal, 2007], [Ferreira, 2007] e [Herbsleb, 1994], dentre eles, o conceito de melhoria contínua, eles ainda possuem uma série de características que dificultam sua utilização no contexto de implantação de testes de software conforme descrito na seção 2.5. Além disso, estes modelos não consideram a

maioria das limitações e restrições inerentes aos programas de melhoria de processos observados nas empresas de desenvolvimento de software [Niazi, 2003], [MCT_01, 2001] e [Santana, 2007] conforme exibido na tabela 4.

Em geral, os modelos apresentados se mostraram complexos para o contexto abordado neste trabalho. É necessário um alto grau de conhecimento nos modelos para que possa escolher quais atividades serão realmente úteis para a organização no programa de melhoria de processo de teste de software. Devido ao grande escopo, esta é uma tarefa bastante trabalhosa e complexa. Além disso, os modelos apresentados são genéricos para programa de melhoria de software, o que torna necessário, muitas vezes, fazer uma adaptação destes modelos para o contexto de testes de software. Esta adaptação pode ter um alto custo para a organização e, muitas vezes, desestimular as pessoas envolvidas, já que demanda muito esforço, tempo e dedicação para sua realização, tornando assim o programa de melhoria de teste inviável.

Tabela 4- Comparação entre os modelos de melhoria

Crítérios	IDEAL	ISO 15504-4	Pro2Pi
Custos de acordo com a realidade da empresa			
Ser de fácil compreensão			
Ser flexível			
Baseados em modelos			
Resultados rápidos			
Modelo Incremental			

Como resultado desta análise, concluímos que se faz necessário elaborar uma abordagem para implantação de testes de software adaptada a partir dos modelos de

melhoria existentes e baseada em um modelo de maturidade específico para testes, aproveitando os pontos fortes dos modelos já disponíveis. O trabalho proposto também deve considerar ações que minimizem os problemas enfrentados neste tipo de iniciativa. Para isso, algumas práticas e princípios de metodologias ágeis foram adicionados ao trabalho proposto. No próximo capítulo, serão apresentadas as metodologias ágeis utilizadas como base para elaboração da abordagem proposta no trabalho.

3

Metodologias Ágeis de Desenvolvimento de Software

Neste capítulo, será mostrada uma visão geral dos conceitos relacionados às metodologias ágeis de desenvolvimento de software. Na Seção 3.1, é apresentada uma breve introdução a metodologias ágeis e seus conceitos relacionados. Na Seção 3.2 é apresentado o Extreme Programming juntamente com seus valores, princípios e práticas. Na seção 3.3, é apresentado o Scrum - outra metodologia de desenvolvimento ágil. Na seção 3.4 são apresentadas as considerações finais do capítulo.

3.1 Introdução

Em 2001, um grupo de profissionais da área de desenvolvimento software decidiu se reunir nos EUA para elaborar alternativas aos processos de desenvolvimento de software pesados e orientados à documentação. Neste encontro, foi observado que embora cada um deles tivesse suas teorias sobre como fazer um projeto de software ter sucesso, todos concordavam que, havia princípios comuns compartilhados entre eles que tinham sido respeitados ao longo dos projetos com sucesso. Com base nisso, foi criado o Manifesto para o Desenvolvimento Ágil de Software [Ste_03, 2007]. O termo Desenvolvimento Ágil identifica metodologias de desenvolvimento que adotam os princípios do manifesto ágil. Estes princípios são os seguintes:

- **Indivíduos e interação entre eles** mais que processos e ferramentas
- **Software em funcionamento** mais que documentação abrangente

- **Colaboração com o cliente** mais que negociação de contratos
- **Responder a mudanças** mais que seguir um plano

O manifesto reconhece o valor dos itens à direita, mas diz que devemos valorizar bem mais os itens à esquerda. Estes princípios servem como orientação básica para qualquer metodologia ágil de desenvolvimento de software.

Nos últimos anos têm surgido diversas metodologias ágeis, tais como: XP (Extreme Programming), FDD (Feature Driven Development), Scrum, ASD (Adaptative Software Development), entre outras.

Dentre todas as metodologias ágeis existentes, duas vêm se destacando em número de adeptos e projetos, que é a XP e o Scrum. Para este trabalho foram selecionadas estas duas metodologias com base nos seguintes critérios:

- Grande quantidade de literatura disponível;
- Ampla utilização em projetos comerciais;
- Muitos relatos de casos de sucesso;
- Possuem todas as características básicas de uma metodologia ágil definida no Manifesto Ágil [Ste_03, 2007].

Nas próximas seções, estas duas metodologias serão descritas em detalhes.

3.2 Extreme Programming

3.2.1 Visão Geral

Extreme Programming (XP) é uma metodologia ágil de desenvolvimento de software, que foi criada nos Estados Unidos em 1996. XP começou a ser desenvolvida a partir de uma experiência em um projeto da empresa DaimlerChrysler. Com o sucesso do XP neste projeto, ela começou a despertar interesse no meio acadêmico e empresarial. Em 1999, Kent Beck publicou o livro “Extreme Programming Explained” [Beck, 1999], onde ele difundiu suas idéias e apresentou os conceitos envolvendo XP.

O sucesso adquirido por XP se deve principalmente aos bons resultados obtidos em projetos, e também devido à sua natureza simples e objetiva por se basear em práticas que já provaram sua eficiência no cenário de desenvolvimento de software [Vasconcelos, 2007]. Criar sistemas com qualidade e que atendam as necessidades do cliente, são alguns dos objetivos do XP. Estes objetivos são alcançados através de um conjunto de valores, princípios e práticas que serão apresentados nas próximas seções.

3.2.2 Valores de XP

Extreme Programming adotou 4 valores que são dimensões nas quais os projetos podem ser melhorados.

Comunicação

Frequentemente grandes problemas são gerados no desenvolvimento de software por causa da comunicação. A comunicação pode causar desentendimentos ou compreensão incorreta no projeto. Essa é uma das razões pelas qual o XP prioriza mecanismos de comunicação.

O XP tem como principal característica forte comunicação entre todos os envolvidos no projeto, ou seja, até mesmo o cliente deve estar no mesmo local que a equipe do projeto para facilitar a comunicação, de forma que os desenvolvedores tenham acesso rápido e freqüente ao especialista no domínio do negócio.

Outro fator que influencia a qualidade da comunicação é a quantidade de pessoas envolvidas. *“Com aumento no tamanho da equipe, torna-se mais difícil para as pessoas saberem o que os outros estão fazendo e como não sobrepor, duplicar ou interferir no trabalho um do outro [Cockburn, 2002, p.151].”* Por esta razão, projetos XP procuram contar com um número reduzido de participantes [Beck, 2005].

Feedback

Requisitos incorretos e incompletos que não refletem a real necessidade dos usuários são um dos principais problemas de um projeto de software. Obter um entendimento

comum dos requisitos entre os usuários e o time de desenvolvimento é uma tarefa difícil e complexa, pois muitas vezes, os próprios usuários não sabem transmiti-los corretamente.

Segundo Brooks [Brooks, 1987], *“nenhuma outra parte do trabalho conceitual é tão difícil quanto estabelecer detalhadamente os requisitos técnicos, incluindo todas as interfaces (...) Nenhuma outra parte é mais difícil de corrigir mais tarde.”*

Para evitar problemas causados pela falta de entendimento ou omissão dos requisitos, o Extreme Programming é organizado em pequenos ciclos que possibilitam aos usuários verificar o funcionamento das funcionalidades desenvolvidas em um curto espaço de tempo. O objetivo é que a funcionalidade seja apresentada para o usuário o mais cedo possível de forma que ele possa estar detectando falhas o quanto antes no ciclo de vida do produto, para que a correção seja menos custosa.

Simplicidade

Para que uma equipe de desenvolvimento possa trabalhar com iterações curtas, é necessário que ela seja capaz de receber um pequeno escopo de funcionalidades no início de cada iteração e implementá-las completamente dentro de um curto período de tempo. Todo o esforço é concentrado apenas na implementação do que foi pedido pelo cliente de modo que qualquer outra funcionalidade nova ou generalizações não solicitadas pelo cliente devem estar fora do escopo da iteração.

Além disso, equipes XP se baseiam no princípio de que “funcionalidades extras adicionam complexidade e não flexibilidade [Poppendieck, 2003].” Sendo assim, qualquer mudança não solicitada ou prevista pelo cliente não deve ser levada em consideração.

Coragem

A coragem de um time é revelada no modo como ele responde ao estresse. XP foi criada para permitir que o time absorva mudanças durante o projeto. Para isso, ela determina que o time do projeto deva ter coragem o suficiente para entender o estado do projeto,

responder às mudanças solicitadas e entregar funcionalidades ao cliente que continuam possuindo valores agregados ao negócio.

3.2.3 Princípios Básicos

Os princípios básicos de Extreme Programming servem para guiar o desenvolvimento de software. Eles procuram classificar os valores dentro do contexto de desenvolvimento. Os quatorze princípios definidos são: auto-semelhança, benefício mútuo, diversidade, economia, falha, fluidez, humanismo, melhoria, oportunidade, passos de bebê, qualidade, redundância, reflexão, responsabilidade aceita. Estes princípios existem para servir de ponte entre valores e práticas servindo como guias que se aplicam a um domínio específico [Ste_01]. Cada princípio é suportado e inserido em um conjunto de práticas definidas no XP que serão apresentadas na seção 3.2.4.

3.2.4 Práticas de XP

Extreme Programming possui doze práticas básicas que compõe o núcleo principal da metodologia. Estas práticas foram elaboradas a partir dos valores e princípios apresentados na seção 3.3.2. Há uma complementaridade muito grande entre as práticas definidas para XP, pois os pontos fracos de uma são superados pelos pontos fortes de outras. Desta maneira as práticas propostas por XP são complementares. Abaixo elas são apresentadas [XP, 2007]:

- **Jogo de Planejamento** (*Planning Game*): As iterações são planejadas semanalmente. Cada iteração possui duração de 1 à 3 semanas. No início de cada semana, as funcionalidades são priorizadas pelos desenvolvedores e cliente. No XP, o escopo é negociável e reavaliado semanalmente. A cada semana, novas funcionalidades são entregues ao cliente completamente testadas, prontas para irem para produção.
- **Pequenas Versões** (*Small Releases*): É necessário que o time de desenvolvimento esteja sempre entregando pequenas versões do sistema para o

cliente. Isso é importante para que o time rapidamente receba feedback do cliente em relação as funcionalidades desenvolvidas. As versões chegam a ser ainda menores que as produzidas por outras metodologias incrementais.

- **Metáfora** (*Metaphor*): É necessário alterar o vocabulário utilizado para um vocabulário que seja simples e mais próximo da realidade do cliente, facilitando assim a comunicação entre o time de desenvolvimento e o cliente do projeto.
- **Projeto Simples** (*Simple Design*): Um projeto simples sempre leva menos tempo do que um projeto complexo. Mantenha as coisas o mais simples possível por quanto tempo for possível, nunca adicionando funcionalidades que não foram previstas.
- **Time Inteiro** (*Whole Team*): O cliente e a equipe de desenvolvimento fazem parte do mesmo time.
- **Testes de Aceitação** (*Customer Tests*): Durante a iteração, o cliente especifica cenários para a funcionalidade que está sendo desenvolvida. O cliente é responsável por verificar se a funcionalidade foi desenvolvida corretamente. Os testes de aceitação também são utilizados como testes de regressão.
- **Ritmo Sustentável** (*Sustainable Pace*): Trabalhar com qualidade, buscando ter ritmo de trabalho saudável (40 horas/semana, 8 horas/dia), sem horas extras. Horas extras são permitidas quando trouxerem produtividade para a execução do projeto.
- **Reuniões em pé** (*Stand-up Meeting*): Reuniões que têm como objetivo melhorar a comunicação entre todos do time focando nas tarefas que estão sendo realizadas, as tarefas que serão realizadas pela equipe e os problemas que estão impedindo a conclusão das tarefas. São reuniões rápidas em pé para não perder o foco no assunto.
- **Posse Coletiva** (*Collective Ownership*): Posse coletiva do código encoraja todos a contribuir com novas idéias em todos os segmentos do projeto. Qualquer pessoa pode mudar o código adicionando funcionalidade, corrigindo *bugs* ou fazendo *refactoring*. É importante que todos conheçam todas as partes do sistema.

- **Programação em Pares** (*Pair Programming*): Todo código que será incluído na entrega é criado por duas pessoas trabalhando juntas em um único computador. Esta forma de trabalho irá adicionar mais funcionalidade com mais qualidade do que duas pessoas trabalhando separadamente. Isso também aumenta a qualidade do código gerado.
- **Padrões de Codificação** (*Coding Standards*): O código deve ser formatado de acordo com os padrões estabelecidos pela equipe de desenvolvimento. Um padrão ajuda a manter o código consistente e fácil para o time inteiro entender e refatorar.
- **Desenvolvimento Orientado a Testes** (*Test Driven Development*): Os testes unitários devem ser criados antes mesmo do código ser desenvolvido. O código desenvolvido deve passar nos testes unitários elaborados.
- **Refatoração** (*Refactoring*): É necessário para manter o código simples e previne erros e códigos complexos. Além disso, o *refactoring* mantém o código limpo e simples, permitindo a melhoria contínua da programação.
- **Integração Contínua** (*Continuous Integration*): A integração contínua é necessária para prevenir conflitos e a possibilidade de erros no código fonte. Sempre que uma nova funcionalidade for finalizada, é necessário que ela seja integrada com a versão atual do sistema.

3.3 SCRUM

Scrum é uma metodologia para desenvolvimento e/ou gerenciamento de produtos e projetos de software. Baseado nas idéias e pesquisas desenvolvidas por Ken Schwaber e Jeff Sutherland em parceria com o Centro de Pesquisas Avançadas Dupont, Scrum foi elaborado e apresentado para a OMG em 1995. As principais características desta metodologia são [Schwaber, 2004]:

- É uma metodologia ágil para gerenciar e controlar o desenvolvimento de software;
- É uma metodologia para desenvolvimento de sistemas e produtos iterativa e incremental ideal para projetos onde há mudanças de requisitos rapidamente e constantemente;
- É uma metodologia que controla o conflito de interesses e necessidades;
- É um modo de melhorar a comunicação e maximizar a cooperação;
- É um modo de detectar e remover qualquer impedimento que possa atrapalhar o desenvolvimento e entrega do produto;
- É um modo para maximizar a produtividade.

Scrum é composto por um conjunto de práticas interrelacionadas que otimizam o ambiente de desenvolvimento, reduz o *overhead* organizacional, e valida os requisitos com protótipos iterativos. Funcionalidades usáveis do produto são entregues em no máximo 30 dias mesmo quando utilizam-se tecnologias instáveis.

Muitas organizações têm tido sucesso usando Scrum para gerenciar e controlar o trabalho em projetos, alcançando melhorias significativas de produtividade [Rising, 2000], [Kniberg, 2006]. O Scrum possui valores, práticas e regras em um framework de desenvolvimento que pode ser rapidamente implementado e repetido.

3.3.1 O Processo

O processo Scrum é descrito pela figura 6. Os projetos possuem uma lista de requisitos (tipo de sistema, itens planejados, tipo de aplicação, ambiente de desenvolvimento, considerações dos usuários, etc). O Scrum armazena requisitos em Backlog do produto, estes requisitos não precisam ser precisos nem descritos completamente. Como em muitos projetos, os requisitos são levantados a partir das expectativas dos usuários ou do negócio. O product owner, responsável por focar no retorno do investimento e pelo backlog, prioriza o backlog do produto: itens de importância para projeto/negócio e

itens que adicionam valores imediatos e significantes ao negócio devem ser colocados no topo da lista.

O time do projeto é responsável por realizar efetivamente o trabalho e criar um Backlog da Sprint: isto compreende os itens de backlog do produto que podem ser finalizados dentro de um período de no máximo 30 dias. O time do projeto pode contactar o Product Owner e outros para expandir os itens no Backlog da Sprint. Depois de decorridos 30 dias, o time deve ter um produto entregável.

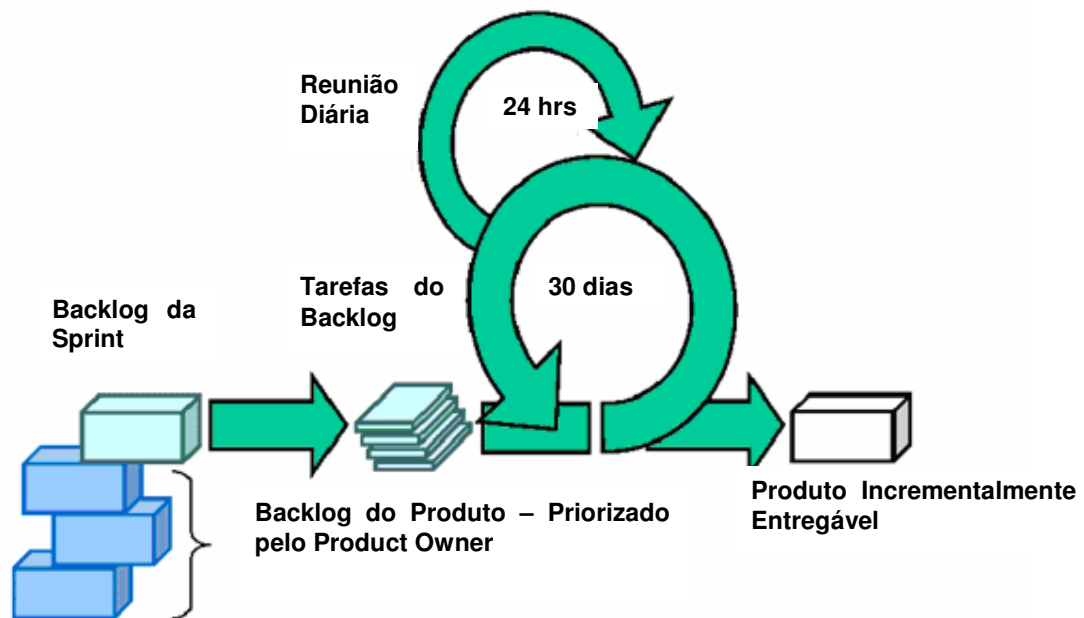


Figura 6 - Processo do Scrum

3.3.2 Regras do SCRUM

Existem valores específicos que devem ser seguidos para obter sucesso no Scrum. Estes valores guiam o seu processo e serão apresentados a seguir [Schwaber, 2004]:

Reunião de Planejamento da SPRINT

A reunião de planejamento da Sprint tem duração de 8 horas e consiste de duas partes com duração de 4 horas cada. A primeira parte consiste em selecionar o backlog do produto; a segunda parte consiste em preparar o backlog da Sprint.

- Os participantes da reunião de planejamento da Sprint são o ScrumMaster, responsável por gerenciar o processo, o Product Owner e o Time. Outras pessoas podem ser convidadas para fornecer informações ou conselhos sobre o domínio do negócio ou da tecnologia.
- O objetivo desta primeira parte da reunião é selecionar os itens do backlog do produto que farão parte de uma funcionalidade a ser entregue. O time irá apresentar esta funcionalidade ao Product Owner e stakeholders na reunião de revisão ao final da Sprint. Esta primeira parte da reunião permite que todos do time possam analisar o backlog do produto. Uma análise mais detalhada do backlog do produto é feita durante a Sprint.
- A segunda parte da reunião acontece logo depois da primeira e também tem duração de 4 horas. A entrada para a segunda parte da reunião é uma lista chamada backlog da Sprint com tarefas e estimativas que o time estará trabalhando para desenvolver a funcionalidade. O product owner deve estar disponível para responder as questões do time em relação ao backlog do produto.

Reunião Diária

A reunião diária do Scrum possui duração de 15 minutos podendo variar de acordo com o número de membros do time.

- As reuniões diárias são sempre no mesmo local e no mesmo horário todos os dias. Todos os membros do time devem participar das reuniões, e se por algum motivo não for possível alguém participar pessoalmente, ele deve participar por telefone ou deve pedir que um outro membro do time participe da reunião por ele.

- Cada membro do time deve responder somente três questões:
 - O que foi feito desde a última reunião diária?
 - O que será feito até a próxima reunião diária?
 - Há algum impedimento para realizar o trabalho?

Os membros do time não devem discutir problemas, discordar das respostas durante a reunião. O ScrumMaster é responsável por gerenciar a reunião não permitindo que ela ultrapasse o tempo estabelecido e não perca o foco.

Sprint

Cada Sprint tem duração de, no máximo, 30 dias consecutivos no calendário. Durante a Sprint, o time deve construir um produto tangível para o product owner e os stakeholders. Esta também é a duração máxima para que muitos stakeholders esperem sem perder o interesse no progresso do time e sem duvidar que o time esteja fazendo alguma coisa que será significativa para ele. O time se compromete com o backlog de produto durante a reunião de planejamento da Sprint [Schwaber, 2004].

- Se a Sprint se tornar inviável, o ScrumMaster pode abruptamente terminar a Sprint e iniciar uma nova reunião de planejamento para iniciar uma nova Sprint. O ScrumMaster pode realizar esta mudança se for requisitada pelo time ou Product Owner. A Sprint pode ser tornar inviável se a tecnologia não é adequada, se as condições de negócios mudaram e a Sprint não vai mais agregar valor ao negócio, ou se o time sofrer interferência externa durante a Sprint.
- Se o time sentir-se incapaz de finalizar todo o backlog da Sprint, o Product Owner pode ser consultado e alguns itens removidos da Sprint atual. Se muitos itens forem removidos e a Sprint perdeu seu valor e significado, o ScrumMaster pode terminar abruptamente a Sprint, conforme mencionado acima.

Reunião de Revisão da Sprint

A reunião de revisão da Sprint tem duração de 4 horas. O time não deve levar mais de uma hora se preparando para esta reunião. O objetivo dela é que o time apresente ao

Product Owner e aos stakeholders as funcionalidades prontas. Porém, o significado de ‘pronta’ pode variar de organização para organização.

- Artefatos não funcionais não são apresentados exceto quando utilizados para suportar o entendimento da funcionalidade demonstrada.
- A reunião de revisão começa com um membro do time apresentando o objetivo da Sprint, o backlog da Sprint que foi acordado e o backlog finalizado. Ao final da reunião, o product owner discute com os stakeholders e o time mudanças no backlog do produto baseados no *feedback*.

Reunião de Retrospectiva da Sprint

A reunião de retrospectiva da Sprint tem duração de 3 horas. Os participantes desta reunião são: o time, ScrumMaster e o Product Owner, sendo que a presença do Product Owner é opcional. Duas perguntas devem ser respondidas por todos os membros do time:

- O que deu certo durante a última Sprint?
- O que pode ser melhorado para a próxima Sprint?

O ScrumMaster registra todas as respostas e o time prioriza quais melhorias em potencial o time gostaria de conversar. Os itens discutidos e acordados podem ser adicionados à próxima Sprint.

3.4 Considerações Finais

O objetivo deste capítulo foi apresentar uma visão geral das metodologias ágeis de desenvolvimento de software escolhidas – XP e Scrum.

Para cada metodologia foram citadas e descritas as suas principais características, valores e práticas. As características, valores e práticas apresentadas neste capítulo serão utilizados e adaptados para abordagem elaborada neste trabalho no

contexto de implantação de testes de software, a fim de torná-la simples, fácil e objetiva, e com isso minimizar os problemas encontrados neste tipo de iniciativa.

No próximo capítulo, o Modelo de Maturidade em Testes (TMM) será apresentado e detalhado. Ele será utilizado como modelo referência em testes para a abordagem proposta e para os modelos de documentos elaborados, fazendo com que a abordagem esteja alinhada a um dos principais modelos reconhecido internacionalmente na área de testes.

4

Modelo de Maturidade em Testes

Este capítulo descreve em detalhes o modelo de maturidade em testes - TMM. A seção 4.1 apresenta uma introdução ao cenário atual de testes de software e aos modelos de maturidade em testes. A seção 4.2 apresenta uma visão geral do TMM. A seção 4.3 apresenta os níveis de maturidade e as áreas de processo contidas em cada nível do TMM. A seção 4.4 apresenta o modelo de avaliação do TMM e seus componentes. A seção 4.5 apresenta uma análise das limitações e dificuldades encontradas no TMM. A seção 4.6 apresenta as considerações finais do capítulo.

4.1 Introdução

Sistemas de software têm se tornado cada vez mais importantes para a sociedade tanto economicamente quanto socialmente. Por causa disso e pela alta competitividade do mercado, a qualidade dos produtos entregues deve ser cada vez mais alta. Para garantir a qualidade do produto desenvolvido é importante que sejam realizados testes de software [Burnstein, 2003].

Para a realização de testes de software de modo efetivo, é necessário que um processo de teste seja estabelecido na organização. Há uma tendência mundial em utilizar o conceito de maturidade para a implantação de processos. Nesta linha existe o CMMI [CMMI, 2001], ISO 15504 [ISO/IEC, 2005], MPS.BR [MPS.BR, 2006] entre outros. Embora estes modelos sejam reconhecidos e utilizados mundialmente, eles não endereçam adequadamente as questões de teste e não possuem o conceito de maturidade

de processo de teste bem definido. Para suprir esta carência, diversos modelos de maturidade de testes foram definidos nos últimos anos, entre os mais conhecidos estão: TPI (Test Process Improvement) [Koomen, 1999], TIM (Test Improvement Model) [Ericson, 1996] e o TMM (Test Maturity Model) [Burnstein, 2003].

Para este trabalho foi escolhido o TMM como modelo de referência para a abordagem desenvolvida por apresentar as seguintes vantagens em relação aos outros modelos:

- Possui maior quantidade de literatura/material disponível;
- Sua estrutura é compatível com o CMMI, modelo que é amplamente utilizado nas empresas;
- Modelo focado em testes mais conhecido no mercado.

Neste capítulo será apresentado e detalhado o Modelo de Maturidade de Teste (TMM), o qual servirá como referência para o desenvolvimento da Abordagem elaborada neste trabalho.

4.2 Visão Geral do TMM

O modelo de maturidade de testes (TMM – Test Maturity Model) foi desenvolvido pelo Instituto de Tecnologia de Illinois com o objetivo de servir como um guia para melhoria de processo e avaliação de processo de teste complementando o modelo CMM¹ [CMM, 2008]. Como o CMM, o TMM também utiliza os conceitos de níveis de maturidade para avaliação e melhoria de processo. Áreas de processos, objetivos dos níveis e práticas também são identificados.

Para a definição de níveis de maturidade, o modelo evolucionário de teste [Gelperin, 1988] foi utilizado como ponto de partida. O modelo evolucionário de testes possui as fases de testes que a organização irá realizar, desde o *debugging* até a fase de

¹ Modelo substituído pelo CMMI.

prevenção de falhas. Além do modelo evolucionário, várias boas práticas da indústria contribuíram para a elaboração do TMM [Burnstein, 2003].

4.3 Níveis de Maturidade

A estrutura do TMM é baseada no CMM e na versão estagiada do seu sucessor CMMI [CMMI, 2001]. O TMM possui cinco níveis de maturidade e para cada nível de maturidade um número de áreas de processos é definido.

Os cinco níveis do TMM irão ajudar a organização a determinar a maturidade do seu processo de teste e definir os passos de melhoria necessários para alcançar um nível mais alto de maturidade de teste.

Os cinco níveis de maturidade e as áreas de processos relacionadas do TMM são:

Nível 1: Inicial

Neste nível, o processo de teste é caótico, indefinido e considerado parte do *debugging*. O objetivo dos testes neste nível é mostrar que o software funciona sem problemas graves. O produto de software é entregue sem visibilidade adequada em relação à qualidade e aos riscos. O produto geralmente não atende as necessidades dos usuários, não é estável e é muito lento. Dentro da área de testes há falta de recursos, ferramentas e testadores treinados. Não existem áreas de processos identificadas para este nível.

Nível 2: Definição

No nível 2, teste é um processo definido e claramente separado do *debugging*. Neste nível, planos são definidos contendo uma estratégia de teste, há derivação dos casos de testes a partir da especificação de requisitos e técnicas formais para elaborar testes são aplicadas. Porém, teste ainda começa relativamente tarde no ciclo de vida de

desenvolvimento. O principal objetivo do teste é verificar que o software satisfaz os requisitos especificados.

As áreas de processo deste nível são: Desenvolver objetivos de testes e *debugging*, Iniciar um processo de Planejamento de Testes, Institucionalizar técnicas e métodos básicos para teste. Estas áreas de processo são descritas a seguir.

Desenvolver objetivos de testes e debugging

O objetivo desta área de processo é diferenciar claramente o processo de teste e *debugging*. Os objetivos, tarefas, atividades e ferramentas para cada um devem ser identificados. Responsabilidades para cada um devem ser atribuídas. Políticas devem ser estabelecidas pela gerência para institucionalizar os processos.

Iniciar um Processo de Planejamento de Testes

O objetivo desta área de processo é estabelecer um processo de planejamento de teste. O planejamento de teste envolve estabelecer objetivos de teste, analisar riscos, estabelecer estratégias, desenvolver especificações do projeto de teste. O plano de teste deve endereçar os recursos, custos e as responsabilidades para os testes unitários, de integração, sistema e de aceitação.

Institucionalizar técnicas e métodos básicos para teste

O objetivo desta área de processo é melhorar a capacidade do processo de teste aplicando técnicas e métodos básicos. Como e quando estas técnicas e métodos são aplicados, bem como as ferramentas básicas para dar suporte devem estar especificados na política e plano de teste. Algumas técnicas e métodos básicos geralmente utilizados no processo de teste são: estratégias de projeto caixa preta e caixa branca, utilização da matriz de validação de requisitos, divisão de testes em subfases como unitário, integração, sistema e aceitação. Algumas ferramentas que dão suporte ao uso de técnicas e métodos são: analisadores dinâmicos e estáticos, analisadores de cobertura, geradores de dados de teste e ferramentas de verificação de erros.

Nível 3: Integração

No nível 3, teste é completamente integrado ao ciclo de vida do software. O planejamento de testes é realizado nos estágios iniciais do projeto. A estratégia de testes é determinada utilizando técnicas de gerenciamento de risco e baseada em requisitos documentados. Uma organização de testes independentemente existe, bem como um programa de treinamento de testes. Além de verificar que o software satisfaz os requisitos, o objetivo de teste neste nível é provar que o produto não funciona corretamente.

As áreas de processo deste nível são: Organização de Teste, Programa de Treinamento de Teste, Integração Teste ao ciclo de vida do software e Acompanhamento e Controle do Processo de Teste. Estas áreas de processo são descritas a seguir.

Organização de Teste

O objetivo desta área de processo é identificar e organizar um grupo de pessoas com foco em teste. O grupo de teste deve ser responsável pelo planejamento, execução, métricas, base de dados e avaliação de teste. O grupo deve ser responsável também pela manutenção do repositório de defeitos.

Programa de Treinamento

O objetivo desta área de processo do ponto de vista do grupo de teste é assegurar que as pessoas estão aptas para desempenhar as tarefas de teste. O programa formal de treinamento é baseado na política de treinamento da empresa. Os treinamentos devem preparar o time de teste para planejar os testes, para realizar tarefas que envolvem integração de teste ao ciclo de vida do software, para o processo de revisão, e para identificar e priorizar riscos relacionados com os testes.

Teste no ciclo de vida do software

O objetivo desta área de processo é promover a integração das atividades de teste em paralelo com outras atividades do ciclo de vida do software. Uma organização madura

não espera começar as atividades de software até o código ser finalizado. Uma variação do modelo V [Sommerville, 2006] é utilizada por gerentes, testadores e desenvolvedores para guiar as atividades de integração.

Acompanhamento e Controle do Processo de Teste

O objetivo desta área de processo é desenvolver o sistema de acompanhamento e controle para o processo de teste para que desvios do plano de teste possam ser detectados o mais rápido possível. O acompanhamento e controle de teste fornecem também maior visibilidade do processo de teste e possibilita a construção de produtos de software com melhor qualidade.

Nível 4: Gerenciamento e Medição

Teste é um processo totalmente definido, fundamentado e mensurável. Revisões e inspeções são realizadas durante o ciclo de vida do software e são consideradas parte do teste. Os produtos de software são avaliados utilizando critérios para características de qualidade. Casos de teste são coletados, armazenados e gerenciado em uma base de dados central para reuso e testes de regressão. Um programa de medição fornece informação e visibilidade sobre o processo de teste e a qualidade do produto.

As áreas de processo deste nível são: Estabelecer um programa de revisão, Estabelecer um programa de medição e Avaliação de Qualidade de Software. Estas áreas de processo são descritas a seguir.

Estabelecer um Programa de Medição

Revisão é um tipo de técnica que pode ser utilizada para remover defeitos de artefatos de software. Esta área de processo tem o objetivo de estabelecer um programa de revisões que ajuda a organização a identificar, catalogar e remover defeitos dos artefatos de software efetivamente e mais cedo no ciclo de vida do software.

Estabelecer um Programa de Medição

O objetivo desta área de processo é estabelecer um programa de medição que identifique, colete, analise e aplique medição para dar suporte à organização determinar o progresso de teste, avaliar a qualidade e efetividade do processo de teste, avaliar a produtividade da equipe de teste, avaliar os resultados de esforços de melhoria de teste, avaliar a qualidade dos produtos de software produzidos.

Avaliação da Qualidade de Software

O objetivo desta área de processo está relacionado à adequação das questões de qualidade de software ao processo de teste, definir atributos mensuráveis de qualidade, definir objetivos de qualidade para avaliação dos produtos de trabalho.

Nível 5: Otimização

Teste é um processo completamente definido e é possível controlar custos e a efetividade dos testes. Neste nível, os métodos e técnicas são otimizados e há um foco contínuo na melhoria do processo de teste. Um método existe para avaliar e selecionar ferramentas de teste. Ferramentas suportam o processo de teste durante as fases de projeto, execução, regressão e gerenciamento de teste. Neste nível, teste tem o objetivo de prevenir defeitos.

As áreas de processo deste nível são: Prevenção de Defeitos, Controle de Qualidade e Otimização do Processo de Teste. Estas áreas de processo são descritas a seguir.

Prevenção de Defeitos

O objetivo desta área de processo é encorajar a organização a analisar os defeitos encontrados. A organização também é encorajada a utilizar a combinação de análise causal de defeito e o plano de ação para guiar a mudança no processo para que os defeitos possam ser eliminados em produtos futuros. As atividades recomendadas para a prevenção de defeitos incluem monitoramento, análise de defeitos, planejamento de

ações, implementação e monitoramento de ações e treinamento em métodos de prevenção de defeitos.

Controle de Qualidade

O objetivo desta área de processo é desenvolver um conjunto de procedimentos de controle de qualidade e práticas que suportam a entrega de software de alta qualidade e que atendam os requisitos do cliente. Alcançar este objetivo permite que a organização incorpore medições avançadas, técnicas e ferramentas para melhorar a efetividade do processo de teste, reduzindo defeitos do software, melhorando a viabilidade e aumentando a usabilidade.

Otimização do Processo de Teste

O objetivo desta área de processo é promover a melhoria contínua do processo de teste e o reuso do processo. A organização é encorajada a identificar práticas de teste que necessitam ser melhoradas, a implementar as melhorias e acompanhar o progresso. É também encorajada a aplicar o processo de controle das atividades de teste, a continuamente avaliar novas ferramentas de teste e tecnologias para adaptação e a suportar a transferência de tecnologia.

A figura 7 exibe as áreas de processo do TMM divididas por nível de maturidade.



Figura 7 - Áreas de Processo do TMM

4.4 Modelo de Avaliação

O modelo de avaliação do TMM utiliza o TMM como seu modelo de referência e é constituído por três elementos: um programa de treinamento para selecionar e instruir o time de avaliação que irá conduzir a avaliação da maturidade, um método de avaliação que permite a organização se auto-avaliar baseada nas respostas dadas a um questionário e nos dados de entrevistas, e por último, um conjunto de questões relacionadas à maturidade para avaliar a maturidade do processo de teste [Burnstein, 2003].

4.4.1 Seleção do Time de Avaliação e Treinamento

Desempenhar uma avaliação de processo de teste, coletar e interpretar dados, e elaborar relatórios finais não é um processo simples. Este processo é de responsabilidade do time de avaliação. Os membros do time necessitam ser instruídos e motivados. Um candidato para o time de avaliação do TMM deve ter conhecimento do método de avaliação do TMM, ser respeitado na organização, estar motivado a melhorar o processo de teste, ter habilidades em implementar mudanças, e ter muito anos de desenvolvimento/testes (uma média de 7 anos). O time de avaliação deve ter um líder, que necessita ter um alto nível técnico e gerencial, experiências em avaliações TMM e habilidades de comunicação.

O tamanho do time de avaliação do TMM varia de acordo com o escopo da avaliação, experiência do time, o tamanho e complexidade dos projetos da organização a serem avaliados. O tamanho do time sugerido é 4-8 pessoas. O escopo do treinamento do time de avaliação deve incluir os seguintes tópicos:

- Introdução aos modelos de melhorias de processo;
- Uma visão geral do TMM;
- Técnicas de entrevista;

- Planejamento da avaliação;
- Análise dos dados;
- Desenvolvimento do relatório.

Cada um destes tópicos pode ser apresentado em forma de mini-cursos.

4.4.2 O Método de Avaliação

O método de avaliação TMM-AM consiste em uma série de passos que guiam o time de avaliação na auto-avaliação do processo de teste. Os principais objetivos do método de avaliação do TMM são:

- Dar suporte à avaliação do processo de teste e determinação do nível de maturidade do TMM;
- Guiar a organização no desenvolvimento de planos de ações para a melhoria de processo de teste;
- Assegurar que a avaliação é executada com utilização eficiente dos recursos da organização;
- Guiar o time de avaliação na coleta, organização e análise dos dados da avaliação;

Os passos do método da avaliação são exibidos abaixo:

Preparação

Preparação para avaliação TMM requer esforço e coordenação de muitos membros da organização. Há muitas tarefas que precisam ser realizadas nesta etapa. A primeira delas é selecionar os membros do time de avaliação e o líder do time. O líder deve certificar que o time realize os treinamentos conforme descrito na seção anterior. Quando o treinamento é finalizado, o time de avaliação desenvolve um plano de avaliação. O plano deve descrever o escopo da avaliação, as entradas e saídas da avaliação, os custos e prazos estimados, tarefas e responsabilidades, mecanismos de controle, e fatores de

risco que podem impactar a saída da avaliação. Técnicas e ferramentas para avaliação devem ser descritas.

Condução da Avaliação

Neste passo o time de avaliação coleta e armazena as informações das entrevistas de avaliação. Um nível do TMM é determinado para a organização baseado na análise dos dados coletados e na utilização do método de classificação do TMM.

Relatório da Avaliação

Como resultado da avaliação é gerado um relatório do processo de teste, o nível do TMM e um registro de avaliação. O time de avaliação prepara um relatório que resume o estado atual do processo de teste da organização. O relatório deve incluir os seguintes itens:

- Uma tabela de conteúdo;
- Um resumo executivo (incluindo o nível de maturidade);
- Uma lista que contém todas as práticas, sub-práticas e suas notas;
- Um resumo de pontos fortes, fracos e áreas não pontuadas;
- Recomendações de áreas para melhoria e prioridades para o plano de ação.

Analisar o Resultado da Avaliação

Neste passo o time de avaliação, junto com gerentes e engenheiros de qualidade, utiliza o resultado da avaliação para identificar e priorizar os objetivos para a melhoria. Uma abordagem para a priorização é descrita por Homyen [Homyen, 1998]. O processo de priorização começa com a seleção de cinco áreas que são as mais importantes no processo de teste. Para cada área é atribuído um valor de 1 a 10 para representar a sua importância na organização. O próximo passo é quantificar a capacidade atual das áreas selecionadas em uma escala de 0 a 10. A prioridade pode ser calculada subtraindo o primeiro passo do segundo. A que obter maior pontuação é a que tem maior prioridade.

4.4.3 Questionário de Avaliação do TMM

O questionário de avaliação do TMM fornece um framework no qual são coletados e armazenados dados da avaliação e fornece instruções para os avaliadores.

O questionário de avaliação do TMM não é a única fonte de entrada para determinação do nível do TMM da organização. Os dados do questionário devem ser confirmados utilizando as informações coletadas nas entrevistas e apresentações.

O questionário de avaliação consiste de oito partes: instruções para o uso, informações do entrevistado, informações da organização, questões relacionadas com as práticas e sub-práticas do TMM, questões sobre ferramentas de teste, questões de tendências de teste, recomendações para melhoria do questionário e um glossário de termos. A figura 8 exibe todas as entradas e saídas do processo de avaliação.

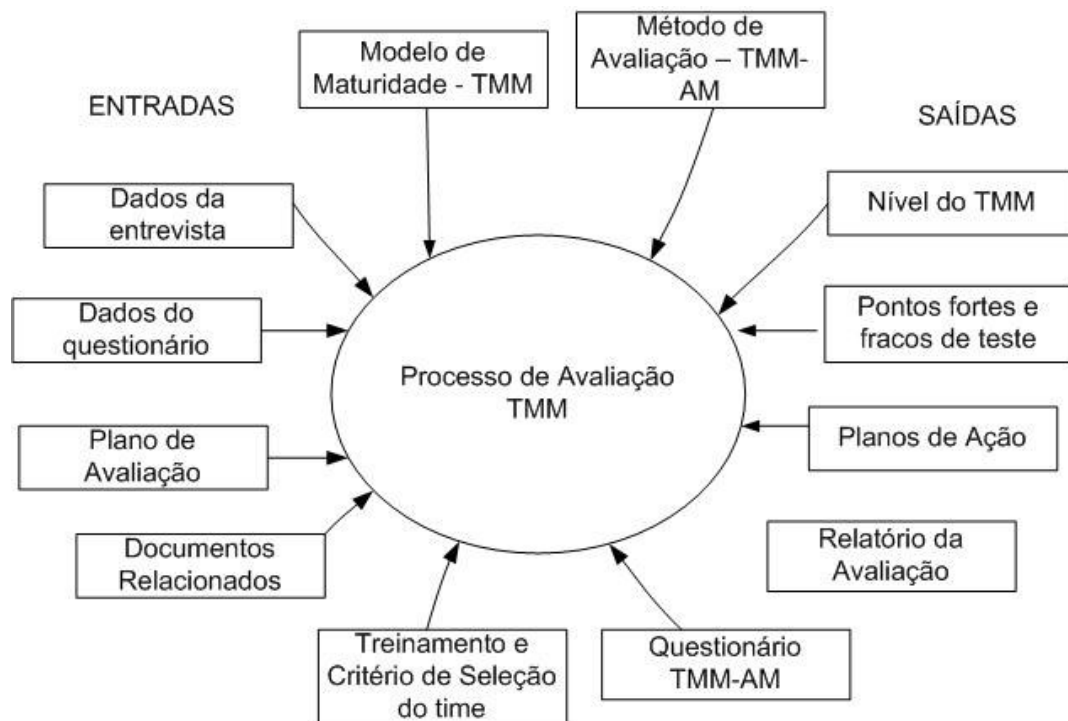


Figura 8 - Modelo de Avaliação do TMM

4.5 Algoritmo de Classificação

O algoritmo para classificação dos processos do TMM começa avaliando se as sub-práticas foram atendidas. Com base nos resultados, avalia-se a satisfação dos objetivos principais de maturidade, e por fim, com base na satisfação desses últimos, é possível identificar se o nível de maturidade foi alcançado. Além disso, o grau de satisfação das sub-práticas permite a identificação de quais áreas no processo apresentam deficiências.

A tabela 5 apresenta a classificação das subpráticas de acordo com as respostas do questionário.

Tabela 5- Nível de satisfação para as sub-práticas

% de respostas = “Sim”	Classificação da satisfação da sub-prática	Grau de Satisfação
> 90	Satisfeito	Muito Alto
70-90	Satisfeito	Alto
50-69	Satisfeito	Médio
30-49	Não Satisfeito	Baixo
<30	Não Satisfeito	Muito Baixo

Uma vez avaliadas as sub-práticas do TMM, é possível avaliar a satisfação dos objetivos principais de maturidade. A satisfação de cada objetivo de maturidade depende dos níveis de satisfação de cada sub-prática relacionada. A tabela 6 exibe como avaliar a satisfação destes objetivos.

Tabela 6 - Classificação de satisfação dos objetivos de maturidade

% de sub-práticas de maturidade	Classificação de satisfação do objetivo de maturidade
≥ 50 satisfeitos	Satisfeito
< 50 satisfeitos	Não Satisfeito
≥ 50 não aplicável	Não Aplicável
≥ 50 não pontuado	Não Pontuado

Para alcançar um nível de maturidade do TMM é necessário que todos os objetivos do nível avaliado sejam satisfeitos.

4.6 Limitações e Dificuldades em utilizar o TMM

Embora o TMM seja um modelo de maturidade bem detalhado e que possui inúmeros benefícios, ele possui algumas limitações e dificuldades em relação à sua utilização no contexto proposto para este trabalho. Com base na análise da autora e experiências de aplicação do modelo presentes na literatura [Veenendaal, 2007], [Karlström, 2004], [Burnstein, 2003] as seguintes limitações foram encontradas:

A primeira dificuldade apresentada é o fato do seu modelo de maturidade ser muito complexo, com muitas práticas e sub-práticas relacionadas o que torna a sua aplicação longa, em média 2 anos para a empresa alcançar o nível 2 do TMM. A segunda dificuldade é em relação ao seu modelo de avaliação, que possui 3 elementos sendo que cada um destes elementos possui uma série de passos que devem ser seguidos para alcançar o sucesso na avaliação, tornando o processo de avaliação muito demorado e custoso. Além disso, o questionário de avaliação é muito extenso e pode levar a uma

má interpretação devido à grande quantidade de termos técnicos utilizados e devido ao fato do questionário estar em inglês, não sendo trivial seu entendimento. No contexto analisado, a maioria das empresas possui funcionários que estão longe de ser especialistas em engenharia ou teste de software. Por isso, é necessário adaptar o vocabulário utilizado no modelo de avaliação para permitir que a empresa possa utilizá-lo em uma auto-avaliação ou necessite apenas de ajudas pontuais.

Outro problema encontrado é que o cálculo do resultado do nível de maturidade obtido pela organização é realizado manualmente não possuindo nenhum auxílio de ferramentas. Sendo assim, este cálculo torna-se demorado, trabalhoso e facilmente suscetível a erros.

E por último, o TMM dá ênfase em “o que fazer” deixando um pouco de lado o “como fazer”, não considerando uma abordagem, estratégia ou processo detalhado de como deve ser desenvolvida a melhoria no processo de testes na organização.

4.7 Considerações Finais

O objetivo deste capítulo foi apresentar o modelo de maturidade em teste TMM escolhido para embasar a abordagem AITS que será apresentada em mais detalhes no próximo capítulo. Para o trabalho proposto, serão considerados apenas os dois primeiros níveis do TMM, já que grande parte das organizações possui baixa maturidade em seus processos de teste.

Para este modelo foram apresentadas suas características, estrutura, níveis de maturidade e áreas de processo. Além disso, foi realizada também uma análise em relação às limitações e restrições do modelo, comprovando assim a necessidade de um modelo de implantação de processo de testes que seja capaz de endereçar estas dificuldades apresentadas no TMM. No próximo capítulo será apresentada a AITS, a abordagem para implantação e melhoria de processo de teste proposta neste trabalho.

5

Abordagem AITS

Este capítulo descreve em detalhes a Abordagem de Implantação de Testes baseada em metodologias ágeis (AITS). O capítulo, além de detalhar a abordagem em si, descrevendo suas fases e atividades, também descreve suas origens e motivações dando uma visão dos benefícios que ela pretende alcançar. A seção 4.1 apresenta uma introdução ao cenário atual. A seção 4.2 apresenta a solução e benefícios da Abordagem de Implantação de Testes no contexto descrito. A seção 4.3 apresenta uma visão geral da abordagem proposta. A seção 4.4 apresenta a abordagem propriamente dita, detalhando suas fases e atividades. A seção 4.5 apresenta as considerações finais do capítulo.

5.1 Introdução

Modelos de maturidade de teste de software se refletem no comportamento das empresas na busca por implantar ou mesmo melhorar o processo de teste utilizado. Embora existam diversos modelos como apresentados nos capítulos anteriores, eles ainda estão longe de abordarem a realidade das empresas que querem começar um programa de implantação ou melhoria de seus processos de teste. Por isso, estas empresas continuam enfrentando uma grande dificuldade com esta disciplina.

Diante deste cenário, o custo/esforço para realizar um programa de melhoria é muito alto, fazendo com que muitas organizações abandonem a idéia de começar um programa de melhoria em testes, causando assim baixa qualidade dos sistemas colocados em produção. Mudar este cenário é uma decisão estratégica para as organizações e é a principal motivação para elaboração da AITS (Abordagem de Implantação de Testes de Software).

5.2 Visão Geral

A abordagem AITS foi elaborada para um contexto caracterizado por organizações intensivas em software com processos de testes de baixa maturidade, que queiram iniciar um ciclo de melhoria de teste o mais rápido possível, investindo poucos recursos e com resultados de curto prazo. O contexto das empresas analisadas apresenta as seguintes características em comum: resistência a mudanças, falta de recursos financeiros, conflito de prioridades entre projetos, dificuldade em medir sucesso, benefícios geralmente obtidos a médio e longo prazo, muitas mudanças culturais e organizacionais em pouco tempo, dificuldade no entendimento dos principais modelos de melhoria de processo de software e dificuldade de adaptação dos modelos para a realidade da empresa.

Esse capítulo descreve as fases e as atividades da abordagem AITS. Seu desenvolvimento foi realizado a partir de estudos realizados nessa área, em modelos disponíveis na literatura apresentados nos capítulos anteriores e na experiência adquirida pela autora atuando com implantação de processos de testes em empresas de software brasileiras. Seu objetivo é contribuir para a melhoria de processos de testes de software, através da implantação de processo de testes por um programa de melhoria, considerando todas as características, limitações e restrições existentes nesta iniciativa.

A abordagem AITS definida é uma adaptação dos modelos de melhoria apresentados no capítulo 2 para o contexto de implantação de processo de teste de software, utilizando o TMM como modelo base de referência para o programa de melhoria e inserindo práticas ágeis presentes no Scrum e XP. Como o contexto estudado para este trabalho é formado por organizações que possuem baixa maturidade nos seus processos de teste, apenas os níveis 2 e 3 do TMM estarão no escopo do trabalho. Além disso, a abordagem AITS considera as melhores práticas dos principais modelos de melhoria.

A utilização destes modelos de apoio se deve ao fato de que eles possuem as seguintes características:

- Modelos reconhecidos internacionalmente;
- Apóiam uma melhoria contínua no processo;
- Existem muitas informações disponíveis sobre eles.

A abordagem definida neste trabalho está organizada em fases, algumas delas sendo executadas apenas uma vez durante todo o programa de melhoria, outras executadas sempre que uma nova iteração for iniciada. Nesta abordagem uma iteração consiste na execução das fases que estão dentro do ciclo conforme exibido na figura 9. Cada iteração terá a duração máxima de 30 dias aproximadamente, conforme definido no Scrum. A adoção desta prática na abordagem se faz necessária para apresentar resultados em um curto espaço de tempo para a organização, motivar os envolvidos com o programa de melhoria e diminuir a resistência em relação às mudanças propostas, já que os benefícios estarão sendo percebidos ao final de cada iteração.

Cada fase é subdividida em atividades que são detalhadas para prover uma orientação efetiva aos responsáveis de como aplicá-las.

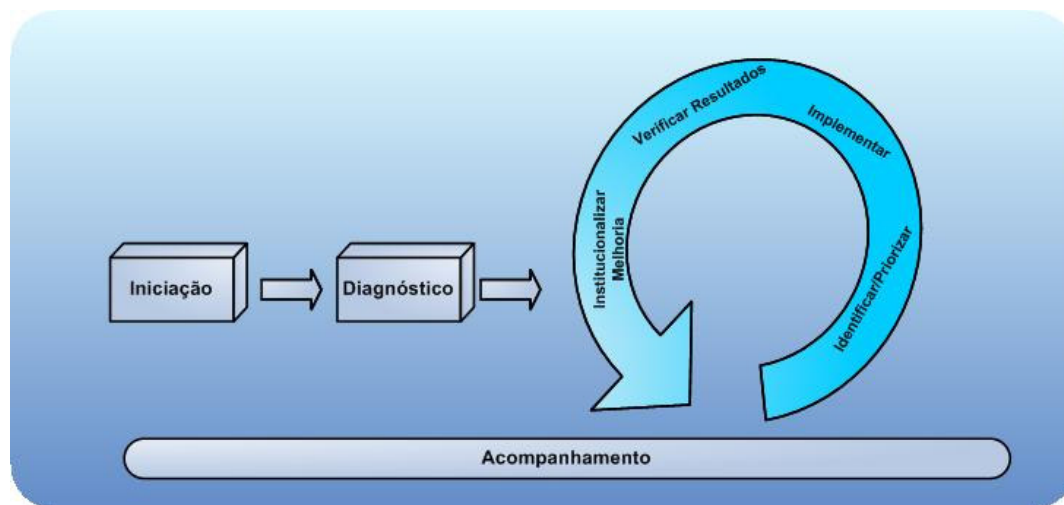


Figura 9 - Fases da abordagem AITS

A figura 9 fornece uma visão geral da abordagem e suas fases: Iniciação, Diagnóstico, Identificar e Priorizar Ações de Melhoria, Implementar Ações de Melhoria, Verificar Resultados e Aprender, Institucionalizar Ações de Melhoria e Acompanhar Programa de Melhoria.

Para a sua aplicação prática, a abordagem necessita que um pequeno time de condutores seja formado para coordenar e conduzir todas as fases durante um programa de melhoria. As pessoas que farão parte do time de condutores deverão possuir preferencialmente os seguintes requisitos para que a abordagem seja aplicada corretamente:

- Experiência em qualidade de software e modelos de melhoria de processos;
- Conhecimento sobre testes de software;
- Boa influência e respeito na organização;
- Boa comunicação;

O time de condutores deve possuir preferencialmente as seguintes características:

- Ter entre 2 e 3 integrantes. É necessário que haja troca de idéias e conhecimentos entre os integrantes do time de condutores. Acredita-se que o time de condutores também não deve ter um número alto de integrantes, 3 seria um número máximo ideal, para que o time não perca o foco do trabalho. Além disso, os times de testes são geralmente pequenos, o que torna desnecessário a formação de um time de condutores com muitas pessoas.
- Cada integrante deverá dedicar pelo menos 50% da sua jornada de trabalho para a condução do programa de melhoria através da abordagem. Isto ocorre pois, na maioria das vezes, os integrantes do time de condutores têm um acúmulo de papéis por causa da falta de recursos financeiros da organização, por isso a abordagem não exige dedicação integral do condutor no programa, mas sua dedicação deve ser pelo menos 50% para que o programa tenha um bom andamento. Não é indicado que nenhum participante do programa de melhoria trabalhe mais do que 8 horas diárias, pois isso pode prejudicar fortemente o seu desempenho no programa. Assim, o princípio de “Ritmo Sustentável” de XP deve ser seguido pelos participantes do programa de melhoria.
- Conhecimentos na Abordagem AITS.

É necessário que os condutores do programa sejam imparciais em relação à execução das fases da AITS para que os resultados não sejam influenciados. A

abordagem recomenda fortemente que um engenheiro de processos faça parte do time de condutores.

Todos os colaboradores da organização que estiverem diretamente ligados à área de testes participarão da aplicação da abordagem no programa de melhoria, sobretudo quando eles estiverem sendo afetados diretamente pelas ações estabelecidas. Por isso, o time de testes não deve ser superior a 12 pessoas, conforme utilizado no XP e no Scrum. A adoção desta prática se faz necessária para maximizar a comunicação entre todos os membros do time e manter o feedback constante entre eles, com o aumento da equipe torna-se difícil manter uma comunicação efetiva.

5.3 Solução e Benefícios

A principal diferença entre a abordagem AITS e os modelos de melhoria de processos existentes atualmente, é o fato da abordagem AITS ser específica para programa de melhoria de testes, fazendo com que todas as suas fases, atividades e templates propostos sejam particulares ao contexto de teste, utilizando como base um modelo de maturidade de testes para tratar particularidades inerentes a este processo.

Além disso, a abordagem AITS foi proposta considerando os problemas comuns inerentes aos programas de melhoria. Para isso, práticas ágeis foram inseridas ao longo das fases e atividades a fim de tornar o programa de melhoria mais simples e ágil seguindo a abordagem proposta. Os conceitos e práticas ágeis que foram utilizados podem ser encontrados ao longo da abordagem proposta.

5.3.1 Visão Geral das Fases da Abordagem

Como visto na figura 9, a abordagem AITS possui sete diferentes fases, das quais 2 delas são realizadas apenas uma vez durante o programa de melhoria. As 4 fases restantes são realizadas diversas vezes, até que todas as melhorias planejadas sejam

implantadas. O conceito de Sprint utilizado no Scrum influenciou fortemente na definição do ciclo da abordagem AITS. Foi escolhido utilizar este conceito, pois ele apresenta grandes benefícios como integração contínua das melhorias, mudanças graduais e diminuição da resistência a mudanças.

Com o objetivo de minimizar os impactos causados pela implantação de um programa de melhoria e otimizar os benefícios trazidos por esta mudança, foi levado em consideração, para definição das atividades de cada fase, a realidade e facilidade com que as organizações serão capazes de incorporar as mudanças no processo e os impactos positivos que os mesmos trarão.

Em cada fase serão listadas as atividades que devem ser realizadas para que o processo implantado siga a abordagem aqui definida. Cada fase possui objetivos gerais que definem o seu propósito. As fases e seus objetivos desta abordagem são os seguintes:

Iniciação

Esta fase tem como objetivo avaliar o contexto e os objetivos de negócios para a realização da mudança na organização, assim como adquirir a decisão e comprometimento dos principais interessados na melhoria. É nesta fase que a alta direção da empresa entende a necessidade de um programa de melhoria de testes e se compromete com o programa. É essencial que nesta fase sejam estabelecidos os seguintes itens para o sucesso do programa de implantação e melhoria de testes:

- Reconhecer e entender os estímulos para a melhoria;
- Definir o contexto do programa;
- Entender os custos e benefícios do programa;
- Identificar os recursos necessários;
- Obter o comprometimento da alta direção.

Diagnóstico

Esta fase tem como objetivo entender o processo atual, as interações organizacionais e como tudo isso contribui com o negócio da organização. As seguintes informações devem ser levantadas nesta fase:

- Pontos fortes e oportunidades de melhoria da organização, que servirão como entrada para a identificação e priorização das ações de melhoria;
- Envolver a alta direção no conhecimento das oportunidades de melhoria para que o programa se torne mais efetivo com seu apoio.

O principal resultado desta fase é um relatório, que contém todos os pontos fortes e oportunidades de melhorias que foram levantados durante as entrevistas do diagnóstico. Uma apresentação com um resumo das informações obtidas é realizada para a alta direção da empresa e para os membros da equipe de teste.

Identificar e Priorizar Ações de Melhoria

Esta fase tem o objetivo de identificar todas as ações necessárias para atingir o estado desejável a partir do estado atual diagnosticado na fase anterior. Uma vez que as ações foram identificadas, é necessário priorizá-las, para que as ações mais urgentes sejam implementadas nas primeiras iterações. O método escolhido para priorização das ações considera 3 critérios principais:

- Nível de maturidade do TMM, onde ações do nível 2 terão maior prioridade do que as do nível 3;
- Importância da ação para a organização;
- Capacidade atual da ação na organização, quanto menor capacidade a ação tiver maior será a prioridade dela.

Nesta fase, também deve ser elaborado um plano de ação considerando as prioridades de cada ação para a organização. Desenvolver o plano de ação é crítico, sendo necessário fornecer um claro direcionamento para as várias ações que deverão ser realizadas em todo o programa. Ele deve ser baseado nos seguintes pontos:

- Nos resultados levantados na fase de Diagnóstico;
- Nas metas de melhoria da organização e;

- Nos recursos disponíveis.

A partir da primeira iteração, esta fase somente será executada caso necessite a identificação de novas ações e/ou re-priorização de ações.

A partir desta fase, todas as fases subsequentes serão realizadas até que todas as ações identificadas sejam implementadas.

Implementar Ações de Melhoria

Nesta fase o plano de ação é executado de acordo com a iteração atual. Utilizando a abordagem iterativa e incremental, as ações são executadas, uma após a outra, conforme definidas no plano, e um projeto piloto é realizado para validar e refinar as melhorias ao final de cada iteração.

Verificar Resultados e Aprender

Esta fase tem o objetivo de verificar os resultados das ações implementadas, para que seja aprimorado o que não foi satisfatório e manter registros de lições aprendidas. É importante nesta fase avaliar os resultados do projeto piloto revisar o que aconteceu durante a iteração, registrar e analisar resultados e lições aprendidas, para que erros não sejam repetidos e o programa ocorra ainda melhor nas próximas iterações.

Institucionalizar a Melhoria

Esta fase tem o objetivo de institucionalizar as ações que obtiveram sucesso no projeto piloto. Nesta fase é importante que:

- Os resultados da iteração sejam exibidos para toda a organização e;
- As melhorias implementadas sejam incorporadas ao processo de teste da organização.

Acompanhar a Melhoria

Esta fase existirá paralelamente a todas as outras, ela tem o objetivo de acompanhar o desempenho de cada fase retornando um *feedback* para os principais interessados, assim como verificar se cada uma das fases atingiu ou não o seu propósito. É necessário que

algumas métricas sugeridas pela AITS sejam estabelecidas para medir se as metas de melhoria da alta direção com o programa foram alcançadas.

5.3.2 Elementos da Abordagem de Implantação

A abordagem é baseada no modelo IDEAL (ver figura 10 – Estrutura da Abordagem de Implantação) que contém os seus elementos estruturais. Esta estrutura contém os seguintes elementos:

- Fase: É um conjunto de atividades relacionadas para atingir um determinado objetivo. Cada fase é composta por:
 - Objetivo: descreve o objetivo geral a ser atingido durante a execução da fase;
 - Atividades: descrevem os passos que devem ser realizados e estão associadas com um determinado objetivo. Um conjunto de atividades é definido para alcançar o objetivo de cada fase. Cada uma destas atividades possui uma descrição, que detalha “o que fazer” e “como fazer”. Cada atividade possui *templates* que refletem e apóiam sua realização. Todos os *templates* definidos neste trabalho poderão ser visualizados inteiramente no apêndice II.
 - Papéis: Descreve as responsabilidades de cada um dos papéis envolvidos em cada fase proposta.

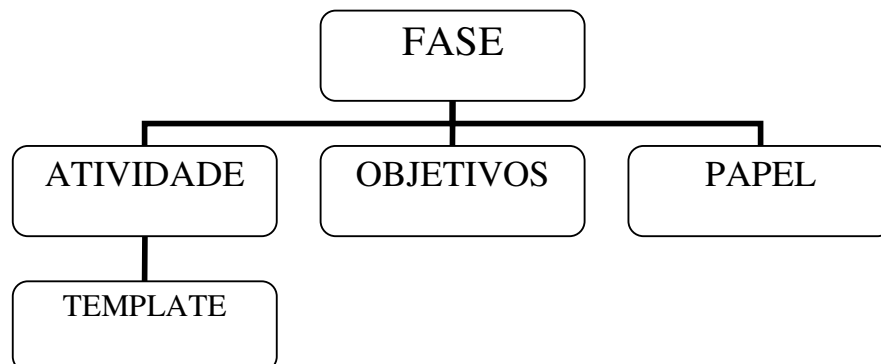


Figura 10 - Estrutura da Abordagem de Implantação

Maiores detalhes sobre cada elemento da AITS serão fornecidos nas seções seguintes.

5.4 Detalhamento das Fases da AITS

A seguir são descritas em detalhes as fases propostas da abordagem de implantação. Cada fase cria uma base para a fase seguinte e só poderá ser considerada como finalizada quando todas as atividades pertencentes a ela forem concluídas.

Nesta seção, para todas as fases iremos detalhar os componentes:

- Objetivos;
- Atividades;
- Templates e;
- Papel.

5.4.1 Fase 1 – Iniciação

Objetivo

Esta fase é iniciada a partir do momento que a alta direção decide iniciar um programa de melhoria na organização. É nesta fase, que a alta direção entende a necessidade, se compromete e define o contexto para o programa de melhoria. Pelo fato de um programa de melhoria causar impactos culturais na forma como a organização trabalha e exigir investimentos, é muito comum observar nas empresas iniciativas voltadas para a qualidade que não são concluídas, gerando grandes frustrações e falta de confiança em ações futuras. Por isso, esta fase é extremamente importante para o sucesso do programa, pois é nela que o apoio e comprometimento da alta direção são estabelecidos. As atividades envolvidas nesta fase são:

Atividades

Obter Comprometimento dos Interessados: Os interessados no programa de implantação de testes devem estar comprometidos com o programa, para isto, é necessário:

- Reconhecer e entender o estímulo para a melhoria: é importante que a alta direção defina quais são os principais motivos para que o programa de melhoria de testes seja iniciado na organização;
- Verificar objetivos de negócio da organização: é importante alinhar os objetivos estratégicos da organização com o programa, para que, ao término dele os objetivos da organização tenham sido atendidos;
- Estabelecer o contexto para a melhoria: identificar as características da organização na qual será realizada a melhoria;
- Obter o comprometimento da alta direção e dos principais interessados: uma vez que os itens acima foram definidos, é importante que os principais interessados se comprometam com o todo o programa de melhoria.

Este comprometimento da alta direção com o programa deve ser registrado através de um documento formal. Este documento pode ser uma ata de reunião, um e-mail ou qualquer outro artefato que contenha o comprometimento formalmente documentado, além dos objetivos estratégicos da organização para com o programa de melhoria de testes. Para esta atividade foi elaborado um template que pode ser utilizado como um documento formal de comprometimento da alta direção.

Kickoff do Projeto: O programa de melhoria deve ser encarado pelos envolvidos como um projeto, que tem início, meio e fim. Todos os envolvidos devem estar cientes qual é o objetivo do programa, quais as dificuldades e benefícios que este programa trará para a empresa. Uma apresentação deve ser realizada para toda a empresa para fornecer informações sobre o programa de melhoria de testes para todos os envolvidos. Esta apresentação deve ser realizada para que todos os envolvidos tenham ciência do

programa de melhoria, sintam-se parte dele e o conheçam como um todo. Com isso, espera-se que todos se sintam beneficiados com o sucesso do programa de melhoria e estejam estimulados para que isso aconteça. Assim, os princípios de “Posse Coletiva” e “Time Inteiro” utilizados no XP são aplicados nesta atividade.

Para facilitar, foi elaborado um template que possui todos os tópicos que uma apresentação de Kickoff de projeto deve ter.

Templates

- Documento de Comprometimento com o programa de melhoria;
- Apresentação de Kickoff do projeto.

Papéis

Os papéis que estarão envolvidos diretamente nesta fase da abordagem são:

- Time de condutores, responsáveis por obter o comprometimento da alta direção com o programa e levantar os objetivos estratégicos da empresa em relação ao programa. Além disso, este time também é responsável por realizar a apresentação de kickoff do projeto;
- Alta direção da empresa: pessoas com poder de decisão, que estejam comprometidos com a abordagem e atuem constantemente no seu acompanhamento, motivação e controle. O papel da alta direção nesta fase é definir os objetivos estratégicos da empresa e se comprometer com o programa de melhoria.

Caso as atividades desta fase não sejam atendidas, ou a alta direção avalie que o programa de melhoria seguindo abordagem AITS é pouco prioritário para a organização, a aplicação da abordagem deve ser cancelada e retornada em uma oportunidade mais favorável. Caso contrário, a aplicação da abordagem no programa de melhoria segue com a execução das próximas fases. A figura 11 ilustra as atividades desta fase.

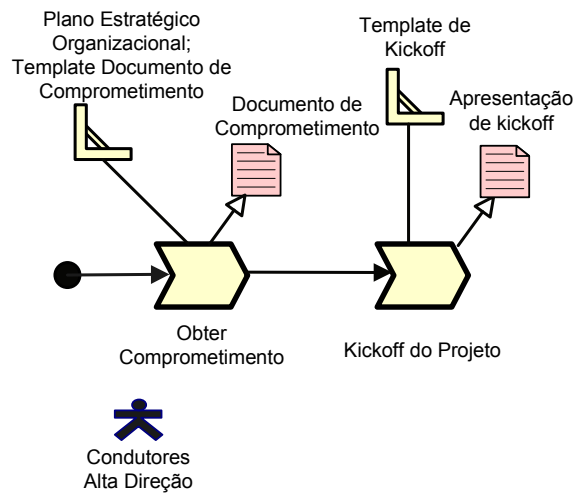


Figura 11 - Atividades da fase de Inicialização

5.4.2 Fase 2 – Diagnóstico

Objetivo

O principal objetivo desta fase é entender o processo de testes atual utilizado pela empresa, assim como as interações organizacionais e como tudo isso contribui com o negócio da organização.

As seguintes informações devem ser coletadas nesta fase:

- Descrição em alto-nível do processo de testes de software da empresa;
- Avaliação do processo de testes para levantar os pontos fortes e oportunidades de melhoria da organização, que servirão como entrada para a elaboração do plano de ação;

Após a realização do Diagnóstico, é necessário envolver a alta direção no conhecimento das oportunidades de melhoria para que o programa se torne mais efetivo com seu apoio.

A técnica utilizada para a avaliação do processo de testes de software é a *benchmarking*, baseada nas boas práticas do modelo TMM. Neste caso, a situação atual do processo de testes da organização é comparada aos requisitos definidos no modelo de

referência do TMM níveis 2 e 3, pois estes são os níveis de maturidade do TMM que estão no escopo deste trabalho.

A seguir as atividades desta fase serão apresentadas em maiores detalhes.

Atividades

Realizar Diagnóstico: esta atividade tem como principal objetivo realizar a avaliação da situação e do processo de testes atual da organização. A avaliação é realizada pelo time de condutores, baseada em um checklist elaborado a partir das práticas e sub-práticas definidas no modelo de maturidade em testes TMM para os níveis 2 e 3. Entrevistas com representantes chave das áreas de testes, requisitos e planejamento de projeto são realizadas, e é fortemente recomendado que elas sejam realizadas em diferentes momentos para cada representante. A importância de realizar entrevistas com membros de outras áreas se deve ao fato de que teste é um processo integrado e afetado por diversas áreas dentro de uma organização. Além disso, é muito importante entender como é que outras áreas enxergam a área de teste, para que os problemas sejam mapeados considerando a organização como um todo. Ao final das entrevistas, é necessário que o time de condutores corrobore as informações fornecidas através de documentos, artefatos produzidos e ferramentas utilizadas pelo processo de testes atual, caso existam. A tabela 7 apresenta uma parte do template do checklist utilizado pelos condutores na avaliação.

Tabela 7 - Checklist de Avaliação

Categoria	Questão	Evidências	Grau de Definição	Observações
Desenvolver Objetivos de Teste e Debugging	Existe política organizacional estabelecida para planejar e realizar os testes?		Não se Aplica	
	Existe um processo definido e institucionalizado para planejamento, projeto, execução e análise de teste?		Não se Aplica	
	Testes unitários são planejados?		Não se Aplica	
	Papéis e responsabilidades para as atividades relacionadas a testes unitários estão estabelecidos		Não se Aplica	
	Testes unitários são projetados?		Não se Aplica	
	Testes unitários são executados?		Não se Aplica	
	O resultado dos testes unitários são registrados/avaliados e problemas são corrigidos?		Não se Aplica	
	Os artefatos de testes unitários são gerenciados e controlados		Não se Aplica	
Planejamento de Testes	o planejamento dos testes é documentado e está coerente com o Plano do Projeto?		Não se Aplica	
	Testes são realizados pela equipe de desenvolvimento antes de serem liberados para a equipe de testes? Quais tipos e estágios?		Não se Aplica	
	O plano de testes possui conteúdo apropriado? . Informações existentes no plano: - Estratégia de Teste - Cobertura dos testes - Estágios e Tipos de testes - Produtos de trabalho a serem testados (casos de uso, classes, métodos etc) - Requisitos a serem testados (iterações) - Priorização dos requisitos - Recursos (Equipe, Ferramentas e Ambientes) - Requisitos não testados - Responsáveis pelos testes - Riscos - Critérios de aceitação		Não se Aplica	
	Recursos apropriados (esforço, prazo, ferramentas, ...) são disponibilizados para o planejamento de testes?		Não se Aplica	

Compilar Resultados: A partir das observações resultantes da avaliação é atribuída uma nota a cada pergunta que indica desde o não atendimento a esta pergunta ao completo atendimento. Esta nota é definida através de um consenso entre os condutores. Com isso, é possível compilar os resultados do diagnóstico, identificando os pontos fortes e as oportunidades de melhorias levantados e corroborados nas entrevistas. Com os resultados do diagnóstico, o time de condutores deverá classificar o processo de testes avaliado em um dos dois níveis de maturidade do TMM considerado por este trabalho. Para ajudar o trabalho dos condutores e evitar erros na classificação de maturidade do processo, o próprio checklist de avaliação determina automaticamente² em qual nível do TMM aquele processo se encaixa, seguindo as notas atribuídas. O processo de classificação seguido pelo checklist de avaliação é muito semelhante ao algoritmo de classificação definido do TMM, conforme descrito no capítulo 4. Abaixo é descrito o processo de classificação utilizado pelo checklist de avaliação da abordagem AITS:

- Questões que tiveram as opções “Não se Aplica” ou “Não Sei” selecionadas não serão consideradas para avaliação, ou seja, serão retiradas do cálculo. Estas opções deverão ser utilizadas em casos extremamente particulares não podendo representar mais do que 10% das respostas do *checklist* de avaliação para não comprometerem o resultado da avaliação.
- As opções “Não”, “Parcialmente” e “Sim” correspondem aos seguintes pesos 0, 1 e 2 respectivamente no resultado da avaliação do processo de teste.
- O Resultado Final é dado da seguinte maneira: se o total de pontos adquiridos nas questões para cada área de processo do TMM for maior ou igual a 70% dos pontos válidos, a área de processo avaliada será considerada como satisfeita. Se todas as áreas de processo do nível forem satisfeitas, o processo de teste será considerado Aprovado naquele determinado nível do TMM. Caso contrário, será considerado como Reprovado. Ressaltando que para o processo de testes seja considerado no nível 3 do TMM, será necessário que ele satisfaça todas as áreas de processo dos níveis 2 e 3.

² O checklist automático foi elaborado no excel.

O critério de aprovação do AITS é considerado bastante arrojado se comparado com o TMM-AM. Isto se deve ao fato do checklist possuir um menor número de questões que conseguem atender as sub-práticas do TMM, fazendo com que estas questões tenham maior representatividade e peso frente às sub-práticas que elas representam. Além disso, o checklist de avaliação da AITS possui a opção “Parcialmente”, o que pode elevar a pontuação final da avaliação se considerado com o TMM-AM que não possui esta opção.

Apresentar Diagnóstico: Deve ser elaborado um relatório final da avaliação contendo os resultados do diagnóstico, este relatório é apresentado à alta direção da empresa. Deve ser também realizada uma apresentação contendo um resumo da avaliação para todos os membros diretamente envolvidos com a área de testes e com programa de melhoria. Esta atividade tem como principal objetivo fornecer uma visão geral do processo de teste atual para todos os interessados no programa. É muito importante que esta apresentação seja realizada para estimular a comunicação e o *feedback* contínuo entre os integrantes do programa de melhoria e obter um entendimento comum em relação ao estado atual do processo de teste da organização. Os participantes poderão questionar, esclarecer suas dúvidas e participar ativamente da apresentação. Nesta atividade são aplicados os princípios de Comunicação e *Feedback* presentes no XP.

Templates para o Relatório Final da Avaliação e Apresentação dos Resultados foram elaborados para apoiar a realização desta atividade.

Templates

- Template do *Checklist*: Este template foi elaborado a partir de uma adaptação do questionário de avaliação do TMM-AM níveis 2 e 3. A principal diferença entre os dois documentos é que o checklist da AITS foi customizado para se tornar um artefato mais simples e mais objetivo, possibilitando um melhor entendimento das questões por parte dos entrevistadores e dos entrevistados, tornando assim, a fase de avaliação mais dinâmica e fornecendo uma compreensão mais rápida aos interessados no programa. O *checklist* de avaliação da AITS possui 64 questões que cobrem todos os objetivos estabelecidos nas áreas de processo do TMM

níveis 2 e 3, enquanto o questionário de avaliação apresentado no TMM-AM possui 104 questões para cobrir os mesmos objetivos. Isto se deve ao fato de que algumas questões do TMM-AM não foram consideradas no questionário de avaliação da AITS por não serem relevantes para o contexto definido neste trabalho. Além disso, algumas questões do TMM-AM também foram diluídas ao longo do *checklist* de avaliação da AITS. É importante ressaltar que embora o *checklist* tenha sido adaptado, ele mantém a cobertura em relação a todos os objetivos definidos pelo TMM nos níveis citados, conforme pode ser observado na tabela 24 do Apêndice I.

- Template do Relatório Final da Avaliação;
- Template da Apresentação da Avaliação.

Papéis

Os papéis que estarão diretamente envolvidos nesta fase da abordagem são:

- Time de Condutores, responsáveis pela avaliação do processo e comunicação dos resultados da avaliação.
- Alta direção, responsável pela leitura do relatório da avaliação;
- Analista de Requisitos, Gerente de Configuração, Gerentes de Desenvolvimento e Todo o Time de Testes deverão participar da apresentação dos resultados da avaliação.
- Gerente de Teste e o Time de Testes: responsáveis por participar das entrevistas do diagnóstico.

É importante ressaltar que esta abordagem suporta melhoria de processos de testes até o nível 3 do modelo TMM, portanto se a organização não for classificada nos níveis 1, 2 ou 3 do TMM, o programa de melhoria estará fora do escopo desta abordagem. A figura 12 exhibe o processo definido para esta fase.

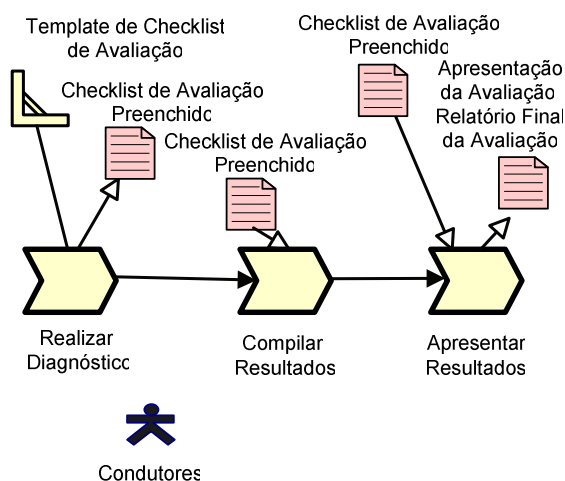


Figura 12 - Atividades da Fase de Diagnóstico

5.4.3 Fase 3 – Identificar e Priorizar Ações de Melhoria

Objetivo

O objetivo desta fase é identificar e priorizar as ações a serem estabelecidas em cada iteração do programa de melhoria de testes da organização. Para isto, os resultados da fase de diagnóstico devem ser considerados. Após a primeira iteração finalizada, as análises e experiências passadas também devem ser observadas.

Atividades

Identificar Ações: Identificar as ações que devem ser realizadas para que os pontos de melhoria levantados sejam resolvidos. Um template com ações padrões foi elaborado baseado em experiências anteriores de melhoria de processos de testes e as ações aderentes às práticas do TMM níveis 2 e 3.

Este template serve como um guia de ações, permitindo que ações sejam adicionadas para atender a realidade de cada empresa. Ações padrões não podem ser substituídas ou removidas, pois todas as ações existentes no documento têm uma relação direta com as práticas do TMM. Sendo assim, é necessário que elas sejam implementadas para que a empresa atinja o nível desejado do TMM. Nesta atividade,

para cada ponto de melhoria identificado na fase de Diagnóstico, uma ou mais ações de melhorias devem ser selecionadas do template de ações padrões para serem priorizadas e posteriormente implementadas.

Priorizar Ações: as ações a serem tomadas deverão ser priorizadas através de critérios que determinarão quais ações devem ser consideradas mais urgentes para a organização e deverão estar nas primeiras iterações do plano de ação. É importante ressaltar que as ações referentes às práticas do TMM nível 2 terão maior prioridade em relação às ações do TMM nível 3 por serem mais básicas e darem suporte para ações do nível 3. Por isso, a pré-priorização e a priorização final só fazem sentido para ações que estão no mesmo nível do TMM.

- É necessário para a priorização das ações, considerar uma análise da importância de cada ação no que se refere às metas da empresa e sua maturidade estimada. Uma pré-priorização é realizada com base na técnica SWOT (*Strengths/Weaknesses/Opportunities/Threats*) [Johnson, Scholes e Sexty, 1989], que explora os pontos fortes e fracos internos de uma organização, suas oportunidades e ameaças. A partir desta análise é atribuído um peso a cada ação que servirá de entrada para a priorização final das ações. A figura 13 exibe o template da técnica SWOT para a pré-priorização.

Pesos das Ações para Priorização			Capacidade Estimada			
			Pontos Fortes		Pontos Fracos	
			<i>Alta</i>	<i>Média</i>	<i>Baixa</i>	<i>Inexistente</i>
Importância (em relação as metas de negócio e melhoria)	Mais Importante	<i>Crítico</i>	6	8	10	12
		<i>Maior</i>	5	7	9	11
	Menos Importante	<i>Menor</i>	1	2	3	4
		<i>Irrelevante</i>	0	0	0	0

Figura 13 - Técnica SWOT para pré-priorização de ações [Weber, 2005]

Uma vez que a pré-priorização foi realizada, é necessário que a priorização final seja feita com base nos seguintes critérios para cada ação:

- Benefícios esperados com sua implementação para a organização;
- Custo de implementação desta ação;
- Interdependência em relação a outras ações;

Um template foi elaborado para avaliar a priorização final das ações considerando todos os critérios citados acima e o resultado da pré-priorização do SWOT. A figura 14 exibe parte deste template.

PRIORIZAÇÃO DAS AÇÕES					
	Peso SWOT	Benefícios	Custo	Interdependência	Ordem de Prioridade
Ação 1					
Ação 2					
Ação 3					
.....					
Ação n					

Figura 14 - Priorização de Ações [Weber, 2005]

A descrição da pontuação permitida nos itens é definida da seguinte forma:

- Benefícios: quanto maior os benefícios que a ação trouxer para a empresa, maior deve ser a pontuação adquirida neste item;
- Custo: quanto maior o custo de implementação da ação, menor deve ser a pontuação adquirida neste item;
- Interdependência: quanto maior a interdependência que a ação tiver em relação a outras ações, menor deve ser a pontuação adquirida neste item.

As ações com maiores pontuações finais são fortes candidatas para fazer parte das primeiras iterações. A pontuação é derivada do somatório da pontuação obtida em cada critério apresentado. Em caso de empate, fica a critério dos condutores escolherem a ação mais prioritária. A priorização das ações se faz necessária para definir o escopo de entrega de cada iteração, priorizando as ações que possuem mais urgência para a organização. Com isso, nesta atividade é feita a aplicação do princípio “Jogo do

Planejamento” utilizado no XP. As ações pertencentes às próximas iterações constituem o backlog do programa de melhoria aplicando o mesmo conceito de backlog utilizado no Scrum.

Elaborar Plano de Ação: Esta atividade consiste em elaborar um plano de ação iterativo e incremental, este plano deve considerar as práticas priorizadas e agrupadas por iterações. As práticas mais prioritárias devem estar nas primeiras iterações do plano. É fortemente recomendado que no máximo 10 ações sejam selecionadas por iteração, a fim de que a equipe de testes não seja sobrecarregada em suas atividades e as mudanças sejam introduzidas gradativamente utilizando o princípio de “Pequenas Versões” presente no XP.

O template do plano de ação definido é dividido em iterações, e as iterações são constituídas por ações. O plano ainda contempla a execução de um projeto piloto ao final de cada iteração para que o time de teste tenha um tempo para colocar em prática o conhecimento adquirido e minimizar a resistência a mudanças. Um piloto final é realizado no final da última iteração planejada para o programa de melhoria consolidar tudo o que foi aprendido durante todo o programa. A figura 15 exibe parte do template de plano de ação contendo ainda datas, marcos, artefatos a serem gerados e recursos envolvidos.

Legendas:				
Status das Ações		Áreas de Processos		
Ações Finalizadas		DEB_TEST	Objetivos de Debugging e Testes	ANA Análise de Teste
Próximas Ações		PLAN	Planejamento de Teste	TEC_MET Técnicas e Métodos de Teste
Ações Em Andamento		PROJ	Projetar Teste	CICLO Ciclo de vida do software
		TREIN	Treinamento Teste	ACOM Acompanhamento de Teste
Ações Planejadas		EXEC	Execução de Teste	ORG Organização de Teste

Processo	Ação	Responsável	Observações	Data Realizado	Data Inicial	Qtd de ReAgend
ITERAÇÃO 1						
PILOTO						
ITERAÇÃO 2						

Figura 15 - Plano de Ação

É importante lembrar que para a primeira iteração todas as atividades desta fase devem ser realizadas obrigatoriamente. Para as iterações consecutivas apenas uma revisão na lista de prioridades deve ser feita. Somente se houver uma mudança muito grande em alguns dos critérios de priorização é que todas as atividades devem ser realizadas novamente. A figura 16 exibe o processo desta fase.

Templates

- Template de Ações Padrões;
- Template de SWOT;
- Template de Priorização;
- Template Plano de Ação.

Papéis

- Time de Condutores e Time de Testes inclusive Gerente de Testes: responsáveis pela identificação e priorização das ações de melhoria de testes.

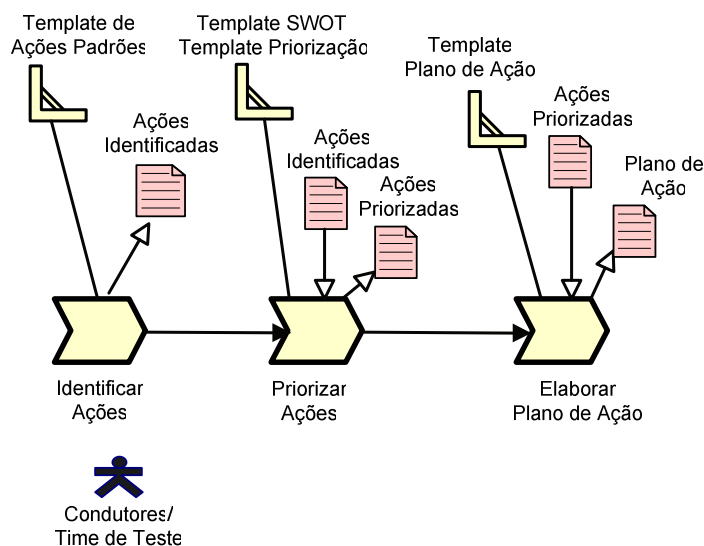


Figura 16 - Atividades da Fase de Planejar e Priorizar Ações

5.4.4 Fase 4 – Implementar Ações de Melhoria

Objetivo

O objetivo desta fase é colocar em prática as ações de melhoria planejadas de acordo com o plano de ação. É nesta fase que solicita-se ao time de testes mudar o modo de como as tarefas de testes estão sendo realizadas. Nesta fase serão realizados pilotos, um ao final de cada iteração a fim de validar e refinar a solução desenvolvida.

Atividades

Executar Ações: Uma vez que o plano de ação foi estabelecido, é necessário implementar as ações planejadas. Nesta fase, deverá ser incluída também a execução do piloto, a fim de avaliar e exercitar o entendimento da equipe em relação às novas ações. Durante o período de execução do piloto, dúvidas que surgirem podem ser esclarecidas, sugestões e observações anotadas (pelos condutores). Os artefatos gerados durante a execução dos pilotos não devem ser reaproveitados para os projetos, afim de que as pessoas se sintam confortáveis para aprender, errar e corrigir. É nesta atividade que os integrantes do time de teste junto com o time de condutores irão testar e darão os aceites

em relação às ações implementadas, eles irão analisar quais ações são passíveis de utilização no dia-a-dia da organização. Com isso, o princípio do XP “Pequenas Versões” é utilizado nesta atividade, pois a cada nova iteração, um conjunto pequeno de ações é disponibilizado aos participantes para que eles possam utilizar e testar as ações através dos projetos pilotos. Outro princípio do XP presente nesta atividade é o princípio “Testes de Aceitação”, uma vez que a equipe já utilizou as ações, eles junto com o time de condutores estão aptos para escolher as ações que serão agregadas ao processo de teste da organização.

Templates

- Não se aplica

Papéis

- Time de Condutores: responsáveis por acompanhar a implementação das ações definidas e dar o aceite em relação às ações implementadas na iteração;
- Time de Testes: responsáveis pela execução das ações definidas e dar o aceite em relação às ações implementadas na iteração.

A figura 17 exibe o processo desta fase.

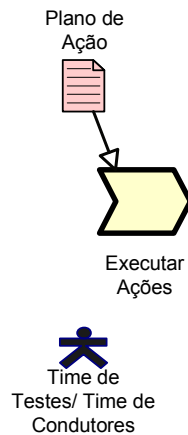


Figura 17 - Atividade da Fase de Implementar Ações de Melhoria

5.4.5 Fase 5 – Verificar Resultados e Aprender

Objetivo

Nesta fase é importante que experiências sejam capturadas e documentadas, bem como os benefícios adquiridos e problemas enfrentados durante a iteração atual. Estas informações devem ser disponibilizadas aos principais envolvidos com o programa de melhoria de testes. Se a iteração atual for a última iteração planejada para o programa de melhoria, os dados obtidos de todas as iterações devem ser divulgados para toda a empresa, mostrando todos os resultados alcançados durante todo o programa.

Atividades

Verificar Resultados: nesta atividade é importante que lições aprendidas sejam coletadas, analisadas e documentadas. A coleta das informações deve ser realizada através de uma reunião com todos os envolvidos na iteração atual do programa de melhoria de testes e questões como: “O que deu certo? E o que poderia ter sido melhor?” devem ser respondidas nesta reunião por cada um dos participantes. O objetivo principal desta atividade é levantar oportunidades de melhoria para a próxima iteração que será iniciada. A coleta desta informação acontece através de uma reunião com todos os membros do time de teste e com todos os que participaram efetivamente durante a iteração. Após o término da reunião, o time de condutores deve documentar os resultados obtidos através de um documento formal. Nesta atividade é utilizado o conceito “Reunião de Retrospectiva da Sprint” definido no Scrum.

Um template foi disponibilizado para facilitar a elaboração do documento de registro de lições aprendidas. Este documento deve ser armazenado em um repositório e consultado sempre que uma nova iteração for iniciada.

Apresentar Resultados: Os resultados da iteração atual devem ser apresentados pelo time de condutores a todos os envolvidos com o programa de melhoria de testes, e também para a alta direção da empresa. As dificuldades levantadas e benefícios da iteração também devem ser apresentados. Se a iteração atual for a última iteração do

programa de melhoria, todos os resultados alcançados durante o programa devem ser apresentados e relacionados com os objetivos de negócios levantados durante a fase de iniciação. Um nível do TMM também deve ser identificado para a empresa com a finalização do programa. Esta atividade é essencial para mostrar a todos os envolvidos, os resultados alcançados por eles na iteração, as dificuldades enfrentadas e etc. Isto auxilia no estabelecimento de uma comunicação forte e contínua entre todos do time, motiva os envolvidos e principalmente a alta direção, pois eles conseguem enxergar os resultados obtidos em um curto espaço de tempo. Com isso, os valores “Comunicação” e “Feedback” do XP estão fortemente presentes nesta atividade.

Um template de apresentação de resultados foi elaborado reunindo todos os itens essenciais que ela deve conter.

Templates

- Template de Lições Aprendidas;
- Template de Apresentação de Resultados.

Papéis

- Time de Condutores;
- Todos os envolvidos com o programa de melhoria
- Alta Direção.

A figura 18 apresenta o processo para esta fase.

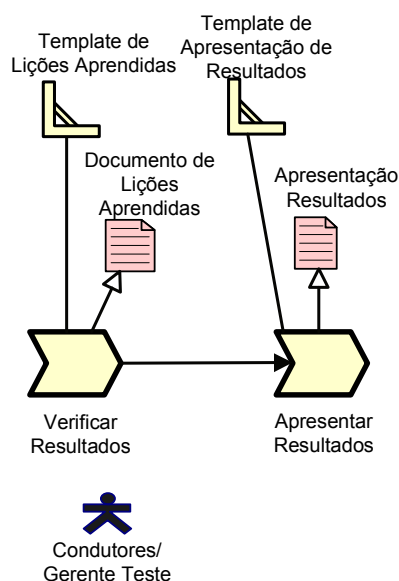


Figura 18 - Atividades da Fase Verificar Resultados e Aprender

5.4.6 Fase 6 – Institucionalizar a Melhoria

Objetivo

O objetivo desta fase é institucionalizar as ações de melhoria que deram resultados positivos nas fases anteriores. Estas ações que deram certo devem ser refletidas no processo de testes da organização, é nesta fase que o processo atual de teste será alterado para contemplar as novas mudanças. É necessário que antes que o processo organizacional seja alterado, todos os envolvidos conheçam as ações que o processo deve refletir.

Atividade

Institucionalizar a Melhoria: todas as melhorias que foram implementadas na iteração e obtiveram resultados positivos, devem ser refletidas no processo de testes da organização. Para isto, os condutores juntamente com o gerente da área de testes devem se reunir e realizar as alterações no processo de testes organizacional. Após a alteração do processo de testes é importante que todos dos times direta ou indiretamente

envolvidos com o processo tomem conhecimento das alterações que foram realizadas. Sendo assim, as ações são agregadas ao processo de teste ao final de cada iteração, inserindo mudanças gradativas e incrementais, diminuindo assim a resistência à mudanças, diminuindo as chances de rejeição às ações de melhoria adotadas, respeitando a cultura da empresa e o tempo que as pessoas levam para se adaptar ao processo modificado. É importante ressaltar que se a iteração atual for a última iteração planejada, o processo de testes deve refletir todas as ações de melhorias implementadas e que obtiveram sucesso durante todo o programa de melhoria de testes. Com isso, o princípio “Integração Contínua” do XP é aplicado nesta atividade.

Templates

- Não se aplica

Papéis

- Time de Condutores;
- Gerente de Testes;

A figura 19 representa o processo para a Fase Institucionalizar Ações de Melhoria

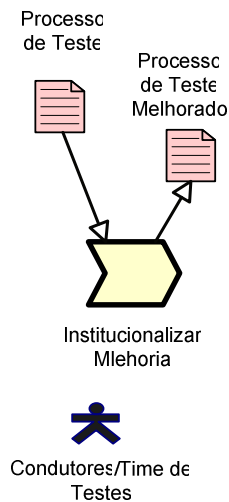


Figura 19 - Atividade da Fase Institucionalizar Ações

5.4.7 Fase 7 – Acompanhamento do Programa de Melhoria

Objetivo

Esta fase tem como objetivo acompanhar a execução e evolução da abordagem, para medir os resultados da implantação de testes trazidos por ela, principalmente em relação aos objetivos da organização. Para isto, dados precisam ser coletados, organizados e analisados através de métricas estabelecidas. Além disso, é necessário um acompanhamento de todas as fases monitorando o cronograma, analisando os resultados finais e experiências de cada fase. É muito importante nesta fase que a alta direção da empresa esteja profundamente envolvida nas atividades.

Atividades

Estabelecer Métricas: Nesta atividade, métricas devem ser identificadas pelos condutores e pela alta direção da organização. Estas métricas devem permitir avaliar se a organização alcançou os objetivos estabelecidos pelo programa.

Um template com métricas básicas foi elaborado para facilitar a realização desta atividade. As métricas contidas no template foram definidas utilizando o paradigma GQM (Goal Question Metric Paradigm), criado por Basili [Basili, 1993] com o objetivo de suportar as organizações na institucionalização de processos de medições, especificamente na identificação de objetivos que serão traduzidos em medições quantitativas. O GQM define orientações para:

- Definir os principais objetivos que serão tratados no programa de medições (Goal);
- Identificar um conjunto de questões que ajudem o atendimento dos objetivos (Question);
- Definição e recuperação de dados que respondam as questões identificadas (Metrics).

Os objetivos que foram utilizados para derivação das métricas padrão do template são objetivos organizacionais comumente encontrados em programas de melhoria. O template definido permite customizações, ou seja, métricas podem ser excluídas, alteradas ou inseridas para cobrir objetivos que eventualmente não tenham sido cobertos pelo template padrão. As métricas selecionadas nesta atividade devem estar de acordo com os objetivos da organização.

Coletar Métricas: nesta atividade as métricas devem ser coletadas para que na atividade seguinte elas sejam analisadas e seus resultados divulgados. Para a coleta, é necessário que seja definida a periodicidade, o responsável, o procedimento de análise e a unidade de medida para cada métrica. É fortemente recomendado que a coleta das métricas seja realizada em pequenos intervalos de tempo, pelo menos uma vez a cada iteração, para que a análise dos dados seja mais precisa. Um template de coleta de métricas foi elaborado a fim de facilitar a execução desta atividade.

Avaliar e Divulgar Métricas: É importante que as métricas que foram coletadas sejam analisadas pelos responsáveis e seus resultados divulgados por toda a organização. Os resultados devem ser comparados com os dados atuais, para avaliar os benefícios que a abordagem trouxe para a organização. Na última iteração, além da avaliação de métricas, é necessário que o checklist de avaliação seja respondido novamente pelas mesmas pessoas, para que uma comparação seja feita entre os resultados anteriores e posteriores à aplicação da abordagem. Neste momento, é necessário que seja avaliado também se os resultados apontam algum desvio em relação aos objetivos organizacionais, e em caso positivo, ações corretivas devem ser implementadas. É indicado que a divulgação seja feita através de boletins internos periódicos para que todos tenham conhecimento do que está sendo feito e alcançado com o programa de melhoria de testes.

Templates

- Templates de Métricas Básicas;

- Template de Plano de Coleta de Métricas.

Papéis

- Alta Direção
- Time de Condutores

A figura 20 representa o processo para a Fase Acompanhamento do Programa de Melhoria.

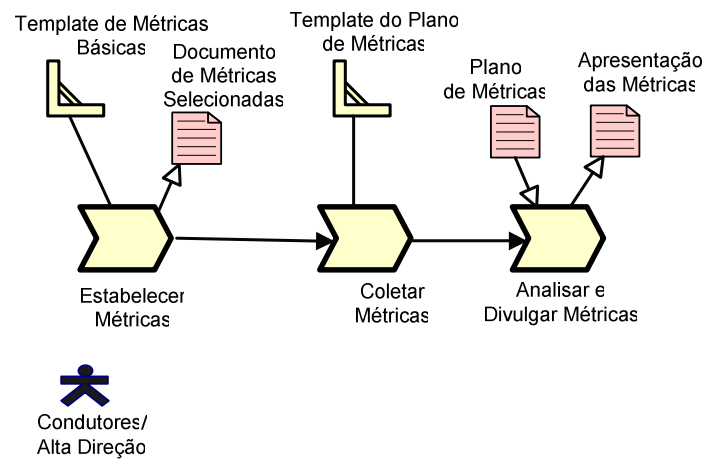


Figura 20 - Atividades da Fase de Acompanhar Programa de Melhoria

5.5 Considerações Finais

Neste capítulo foi apresentada a Abordagem de Implantação de Testes AITS. A abordagem tem como principal objetivo implantar processos de teste de uma forma organizada, madura e ágil, de modo que as empresas encontrem maior facilidade em implementar um programa de melhoria. Para isto, a AITS é uma adaptação dos principais modelos de melhoria, envolvendo conceitos e princípios de metodologias ágeis assim como práticas do modelo de maturidade em testes TMM. Na tabela 8 são

apresentados todos os princípios e conceitos das práticas ágeis que podem ser encontrados nas fases e atividades apresentadas da abordagem AITS.

Tabela 8 - Mapeamento das atividades da AITS com conceitos e princípios ágeis

Prática Ágil	Onde encontrar na AITS	Descrição Fases/Atividades/Estrutura
Sprint	Estrutura ³	As iterações devem ser de no máximo 30 dias. Presente na estrutura da abordagem
Ritmo Sustentável	Estrutura	Define que os membros devem dedicar pelo menos 50% da sua jornada de trabalho para o programa de melhoria, não ultrapassando 8 horas diárias. Presente na estrutura da abordagem
Equipe com no máximo 12 integrantes	Estrutura	O tamanho do time não deve ser maior que 12 pessoas. Presente na estrutura da abordagem.
Posse Coletiva	Atividade	Kickoff do Projeto
Time Inteiro	Atividade	Kickoff do Projeto
Comunicação	Atividades	Apresentar Diagnóstico, Apresentar Resultados
Feedback	Atividades	Apresentar Diagnóstico, Apresentar Resultados

³ Este termo é utilizado para indicar que a prática e/ou princípio está presente na definição da abordagem sem necessariamente estar associado a uma atividade específica.

Prática Ágil	Onde encontrar na AITS	Descrição Fases/Atividades/Estrutura
Jogo do Planejamento	Atividade	Priorizar Ações
Pequenas Versões	Atividade	Executar Ações, Elaborar Plano de Ação
Testes de Aceitação	Atividade	Executar Ações
Reunião de Retrospectiva da Sprint	Atividade	Verificar Resultados
Integração Contínua	Atividade	Institucionalizar Melhoria

Em relação aos modelos de melhoria apresentados neste trabalho, a abordagem AITS apresenta as seguintes diferenças:

- Necessita a criação de somente um grupo específico dentro da organização para a implementação/melhoria do processo de teste, que é o time de condutores. Além disso, este time conta com a participação dos próprios colaboradores da organização. Isto faz com que os custos do programa de melhoria sejam reduzidos, já que não há necessidade de contratação de uma consultoria externa e nem há um alto número de funcionários alocados em atividades do programa de melhoria;
- Incorpora muitas práticas/princípios ágeis que a torna mais simples e ágil, contando com a presença de um conjunto mínimo de atividades necessárias para a realização de um programa de melhoria, com templates que apóiam a realização das atividades e com um vocabulário simples, evitando más interpretações;

- É específica para o contexto de testes unindo os benefícios trazidos pelos principais modelos de melhoria de processo e por um modelo de maturidade em testes – TMM. Conta ainda com os templates elaborados especificamente para o contexto de melhoria de processo de testes, possuindo ao mesmo tempo o “como fazer” e “o que fazer” para melhorar um processo de teste em uma organização;
- Elaborada considerando as principais limitações e dificuldades enfrentadas pelas empresas de software em um programa de melhoria;
- O processo de avaliação contido na abordagem é bastante simplificado se comparado aos processos de avaliação dos modelos de melhoria apresentados. Conta ainda com o apoio de um template que calcula automaticamente o nível do TMM que a organização se encontra baseado nas respostas dadas pelos entrevistados, minimizando assim as chances de erros inerentes a este processo.

Todos os modelos de documentos (templates) citados neste capítulo podem ser visualizados no Apêndice II deste trabalho.

O próximo capítulo irá apresentar os resultados obtidos com a aplicação desta abordagem em duas diferentes situações. O objetivo foi avaliar a abordagem proposta para implantar e/ou melhorar a área de testes das organizações participantes.

6

Aplicação da Abordagem AITS

Este capítulo descreve dois estudos de caso realizados com a aplicação da metodologia proposta nesta dissertação. A seção 6.1 apresenta uma introdução dos estudos de caso realizados. A Seção 6.2 expõe os resultados do primeiro estudo de caso. Na Seção 6.3, são descritos os resultados do segundo estudo de caso realizado. Na seção 6.4 foi realizada uma análise dos benefícios das práticas ágeis presentes na abordagem. As considerações finais sobre o capítulo são apresentadas na seção 6.5.

6.1 Visão Geral

Neste capítulo são apresentadas duas experiências de aplicação da abordagem AITS em empresas de desenvolvimento de software, realizadas no estado de Pernambuco entre 2006 e 2007.

As experiências, baseadas em estudo de caso, iniciaram durante a definição da abordagem AITS, auxiliaram sua definição e foram fundamentais para sua melhoria. Os dados gerados e os problemas enfrentados durante os estudos de caso permitiram analisar sua aplicabilidade no contexto de implantação e melhoria de testes de software e avaliar os resultados alcançados com sua execução.

As seções a seguir descrevem como cada estudo de caso foi realizado e como a implantação e a melhoria do processo de teste foram estabelecidas, descrevendo também a duração e os custos envolvidos.

Os dados apresentados foram coletados de reuniões, entrevistas, questionários, documentos e ferramentas utilizados durante a implantação e melhoria do processo de teste.

6.2 Estudo de Caso 1

O primeiro estudo de caso, também descrito em [Diniz et al, 2007], foi realizado numa empresa voltada para o desenvolvimento de ERP (*Enterprise Resource Planning*) para os mais diversos setores. A empresa conta atualmente com 40 colaboradores, sendo que 7 deles fazem parte do time independente de teste da empresa.

Em julho de 2006, a empresa iniciou um programa de melhoria de processo de teste em parceria com o NEXT [NEXT, 2007], laboratório de testes vinculado ao Softex-Recife [SOFTEX, 2007]. Na mesma época a abordagem AITS estava sendo desenvolvida. Neste estudo de caso, dois colaboradores do NEXT, sendo que a autora era um dos colaboradores, foram alocados ao time de teste da empresa e foram responsáveis por dar orientação na aplicação da abordagem e na implantação do programa de melhoria de testes. O projeto selecionado para a abordagem foi um projeto de manutenção corretiva do principal produto da organização, o qual chamaremos de Produto 1. O produto 1 é considerado o produto principal da empresa e atualmente está implantado em mais de 30 clientes possuindo mais de 500 funcionalidades. Além de atividades de manutenção corretiva, freqüentemente são realizadas customizações no produto para os clientes que o adquirem.

Neste estudo de caso foi utilizada a versão inicial da abordagem AITS, a fim de avaliar sua aplicação e refiná-la com base nos resultados obtidos.

Para iniciar a aplicação da abordagem o grupo de condutores foi estabelecido, sendo formado pelos dois membros do NEXT alocados na empresa e a gerente da equipe de testes. Foi definido também que durante os meses da realização do programa de melhoria, os integrantes do time de condutores e teste dedicariam 50% da sua carga horária para as atividades propostas na abordagem.

Na fase de iniciação, foi obtido o comprometimento entre a alta direção, time de condutores e o time independente de testes através de uma reunião, onde foram definidos dois objetivos da organização em relação ao programa de melhoria de testes:

- Melhorar o processo de teste;
- Aumentar a qualidade do produto.

Os objetivos e o comprometimento da alta direção e de todo o time de teste foram registrados em uma ata de reunião.

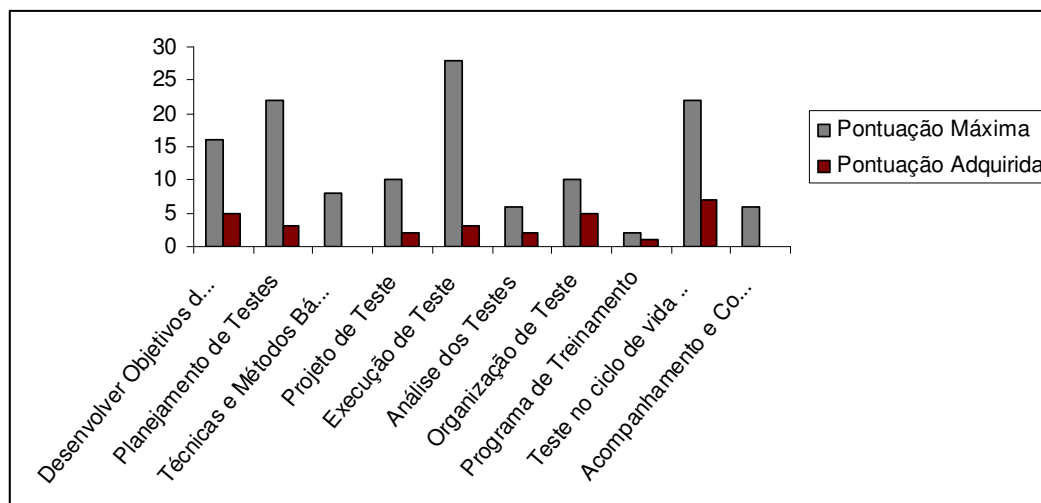
Na fase de diagnóstico, dois membros da equipe de condutores realizaram 4 entrevistas para preenchimento do checklist de avaliação, em diferentes momentos, com membros direta ou indiretamente envolvidos com a área de testes. A primeira entrevista foi realizada com 3 analistas de testes da equipe, a segunda com os 3 analistas de testes restantes, a terceira entrevista foi realizada com a gerente do time de testes e a quarta e última entrevista foi realizada com um analista de requisitos e com o gerente do projeto. Os resultados dos checklists de avaliação das entrevistas foram consolidados pelos condutores em um único checklist de avaliação. Na tabela 27 do Apêndice I são apresentadas as informações consolidadas do checklist de avaliação por área de processo detalhadamente, na tabela 9 é apresentado o resultado final da avaliação, onde pode ser visualizado os pontos adquiridos pelo processo de teste da empresa nos níveis 2 e 3 do TMM em relação ao total de pontos que poderiam ser alcançados pelo níveis do TMM citados. Na tabela 10 é apresentado o gráfico do diagnóstico final por área de processo.

Resultado Final do Diagnóstico

Tabela 9 - Avaliação do Processo de Teste do Estudo de Caso 1

RESULTADO DA AVALIAÇÃO		
PONTOS	Nível 2	Nível 3
Total de Pontos Válidos	90	40
Total de Pontos Adquiridos	15	13
RESULTADO	REPROVADO	REPROVADO

Tabela 10 - Gráfico do Diagnóstico em relação as áreas de processo do Estudo de Caso 1



Após a consolidação dos resultados, uma apresentação foi realizada para os membros da equipe de teste e principalmente para a alta direção. Nesta, reunião foram apresentados os pontos fortes e pontos fracos da área de teste da organização levantados pelo time de condutores durante o diagnóstico. Entre os pontos fortes da organização estavam:

- Investimento da alta direção em melhoria de processos, particularmente na implantação do MPS.BR (Modelo Brasileiro para Melhoria de Processo de Software) nível G;
- Investimento em recursos humanos para a equipe de testes;
- Especificação dos requisitos realizada através de casos de uso;

- Roteiros de testes elaborados contendo situações críticas.

Entre os pontos fracos da organização estavam os seguintes:

- O processo de teste atual não detalha as fases de planejamento, execução, análise de teste e liberação do produto;
- A equipe de teste não foi envolvida na definição do processo, não foi treinada e não utilizam o processo definido;
- A equipe de teste é envolvida apenas no final do ciclo de vida do software;
- O ferramental existente não é suficiente para apoiar todo o processo de teste;
- Percepção de baixa qualidade do produto pela equipe;
- Não existe um mecanismo formal e institucionalizado de comunicação: reuniões periódicas não são realizadas, relatórios formais e periódicos não são produzidos, não está estabelecida a forma de comunicação entre as áreas e a forma de divulgação dos resultados;
- O planejamento dos marcos, liberação de versões e escopo dos produtos não é formalmente estabelecido e comunicado para as equipes afetadas;
- Avaliações de satisfação dos clientes não são realizadas periódica e formalmente;
- Requisitos não-funcionais não estão especificados;
- A equipe de testes não participa de revisões de requisitos dos projetos;
- Não há estimativa de esforço das atividades da equipe de testes;
- Algumas medições sobre qualidade do produto são geradas sob demanda, mas não são amplamente divulgadas na empresa;
- Critérios de aceitação dos produtos não são estabelecidos;
- Não há planejamento de rodadas/suites de testes;
- Casos de Testes não são especificados;

- Os testes realizados pela equipe de testes não são realizados sobre baselines formais (versões estáveis) do produto;
- Testes automáticos não são realizados;
- Não são executados testes não-funcionais;
- Não há um padrão para registro da execução de testes;
- A forma de registrar bugs encontrados não é efetiva, pois não é classificada severidade, prioridade, imagens do bug e não há um controle sob o ciclo de vida do bug;
- Relatórios de análise de testes não são elaborados;
- Bugs encontrados pelo cliente não são analisados para melhoria do processo.

Uma vez que a fase de Diagnóstico foi finalizada, iniciou-se a fase de Planejamento das Ações de Melhoria. Primeiramente, um conjunto de ações foi selecionado baseado nas práticas do nível 2 e 3 do TMM. Esta seleção considerou principalmente as deficiências da organização diagnosticada na avaliação da fase anterior. Logo em seguida, as ações foram priorizadas e o plano de ação iterativo e incremental elaborado. O plano de ação elaborado definiu a realização de quatro iterações conforme exibido na tabela 11. Um planejamento detalhado com datas, marcos, produtos a serem gerados foi elaborado no início de cada iteração e aprovado pela empresa.

Tabela 11 - Plano de Ação

Legendas:						
Status das Ações		Processos				
Ações Finalizadas		DEB_TEST	Objetivos de Debugging e Testes	ANA	Análise de Teste	
Próximas Ações		PLAN	Planejamento de Teste	TEC_MET	Técnicas e Métodos de Teste	
Ações Em Andamento		PROJ	Projetar Teste	CICLO	Ciclo de vida do software	
		TREIN	Treinamento Teste	ACOM	Acompanhamento de Teste	
Ações Planejadas		EXEC	Execução de Teste	ORG	Organização de Teste	
Processo	Ação	Responsável	Observações	Data Inicial	Data Final	Re Agend
PRÁTICAS BÁSICAS DE PROJETAR E AUTOMATIZAR TESTE						
GERAL	Definir como o processo será documentado: como as atividades deverão ser descritas, padrões para templates, padrões para guias, fluxo de atividades.			8/ago	15/ago	
PROJ	Definir ferramenta de gerenciamento de testes			16/ago	18/ago	

PROJ	Definir forma de especificar casos de testes Elaborar guia de especificação de casos de testes Definir forma para priorização de casos de testes Definir critério para definir o procedimento de testes dos casos de testes prioritários Definir critério para definir massa de dados Documentar atividades, templates e guias			18/ago	31/ago	
AUTOM	Definir e documentar critérios para seleção e registros dos casos de testes a serem automatizados Definir processo de automação			31/ago	8/set	
PIL	Piloto: Especificar e automatizar alguns casos de testes			8/set	15/set	
PRÁTICAS BÁSICAS DE EXECUÇÃO E ANÁLISE DE TESTES						
EXEC	Definir como os defeitos/observações serão registrados			25/set	27/set	
PLAN	Definir medições iniciais			28/set	5/out	
ANA	Definir template para relatório de testes			6/out	9/out	
PLANDET	Definir planejamento de suítes e rodadas de testes			10/out	12/out	
GERAL	Definir processo de inspeção Estabelecer inspeções para o Projeto de Testes, Scripts e Relatório			12/out	19/out	

	de Testes					
PIL	Piloto: inspecionar projeto de testes, executar um conjunto de testes, registrar os defeitos como padrões definidos, gerar relatório de testes com medições iniciais, inspecionar relatório de testes			20/out	27/out	
PRÁTICAS AVANÇADAS DE PLANEJAMENTO						
PLAN	Definir método de estimativa de esforço e prazo			6/nov	9/nov	
PLAN	Definir reunião de lições aprendidas, análise de bugs identificados pelo cliente para melhoria interna			10/nov	17/nov	
PLAN	Definir métricas de produtividade e qualidade do produto			20/nov	27/nov	
PIL	Piloto: Realizar piloto com práticas avançadas de planejamento			28/nov	8/dez	
PRÁTICAS AVANÇADAS DE REQUISITOS						
REQ	Estabelecer requisitos não funcionais			Não Realizada		
REQ	Analisar forma de especificação de requisitos: verificar se a mesma está adequada para testes			Não Realizada		

PROJ	Estabelecer casos de testes de requisitos não funcionais – Performance			Não Realizada		
AUTOM	Implementar casos de testes de requisitos não funcionais – Performance			Não Realizada		
REQ	Definir controle de versões de requisitos - baselines de requisitos. Como alterações aos requisitos serão comunicadas ao teste			Não Realizada		
REQ	Definir rastreabilidade de requisitos			Não Realizada		
PIL	Piloto: Realizar piloto com práticas avançadas de Requisitos			Não Realizada		

A fase seguinte, Implementar Ações de Melhorias, foi realizada de forma que toda a equipe de teste estivesse envolvida na execução das ações de melhoria planejadas para a iteração corrente. Ao final de cada iteração, um piloto exercício era realizado com o objetivo de treinar a equipe nas práticas definidas no âmbito da realidade da empresa. Os produtos produzidos durante estes pilotos não foram reaproveitados para os projetos, afim de que as pessoas se sentissem confortáveis para aprender, errar e corrigir. As atividades deste piloto não entraram nos cronogramas dos projetos em andamento.

A fase Implementar Ações de Melhoria foi realizada 3 vezes durante o todo o programa de melhoria. A última iteração planejada no plano de ação não foi realizada, pois a organização estava implantando, junto com uma consultoria especializada, o MPS.BR nível G, o qual requer a definição dos requisitos não funcionais e práticas de

rastreabilidade. Desta forma, a empresa optou por evoluir os testes desta área após a certificação, quando o processo de requisitos já estivesse maduro.

Na fase Verificar Resultados e Aprender, reuniões ao final da iteração eram realizadas com a equipe de testes e membros de equipes afetadas por práticas implantadas naquela iteração. As reuniões tinham duração de aproximadamente duas horas e as seguintes perguntas eram respondidas por todos os participantes:

- O que deu certo durante a iteração?
- O que pode ser melhor para a próxima iteração?

Ao final da reunião, um dos membros do time de condutores reunia e compilava todas as informações para que elas pudessem ser apresentadas para o time de testes.

Ao final de cada iteração o conjunto de práticas que tiveram aceitação positiva pelo time de teste era agregado ao processo de teste da organização. Quando todas as iterações foram finalizadas, todas as práticas estavam agregadas ao processo de teste da organização definido durante o programa de melhoria. Esta fase caracterizou a fase de Institucionalizar a Melhoria.

A fase Acompanhar Melhoria foi realizada durante todo o programa de melhoria. Para melhor acompanhamento das fases, as seguintes métricas exibidas na tabela 12 foram definidas e coletadas:

Tabela 12 - Métricas definidas para acompanhamento do programa

Necessidade da Informação	Objetivo	Métrica
Acompanhar a duração de cada fase	Conhecer o desvio da estimativa de cada fase do programa	Desvio em % por fase em relação à estimativa de prazo
Acompanhar o custo de cada fase	Acompanhar o custo de cada fase do programa	Esforço gasto na execução de cada fase

A coleta destas métricas foi realizada ao longo do programa pelo time de condutores juntamente com a alta direção da empresa. Ao final do programa de melhoria os dados foram consolidados e os resultados analisados. As tabelas 13 e 14 apresentam os dados coletados juntamente com os resultados obtidos para as métricas definidas acima.

Tabela 13 - Duração das fases do programa

Fases	Planejado por Iteração (Média)	Realizado por Iteração (Média)	Desvio
Inicialização	5 dias	7 dias	40%
Diagnóstico	2 dias	2 dias	0%
Identificar e Priorizar Melhoria	3 dias	4 dias	33%
Implementar Ações	20 dias	30 dias	50%
Verificar Resultados	2 dias	2 dias	0%
Institucionalizar Melhoria	5 dias	5 dias	0%
Acompanhamento do Programa	Durante execução de	Durante execução	N/A

	outras fases	de outras fases	
--	--------------	-----------------	--

Tabela 14 - Esforço gasto em cada fase do programa

	Horas gastas pelo Time de Condutores	Horas gastas pela Alta direção, time de teste, membros de outros times
Inicialização	28	28
Diagnóstico	8	6
Identificar e Priorizar Melhoria	16	24
Implementar Ações	180	360
Verificar Resultados	24	6
Institucionalizar Melhoria	10	20
Acompanhamento do Programa	15	30
Total de Horas Gastas	755	

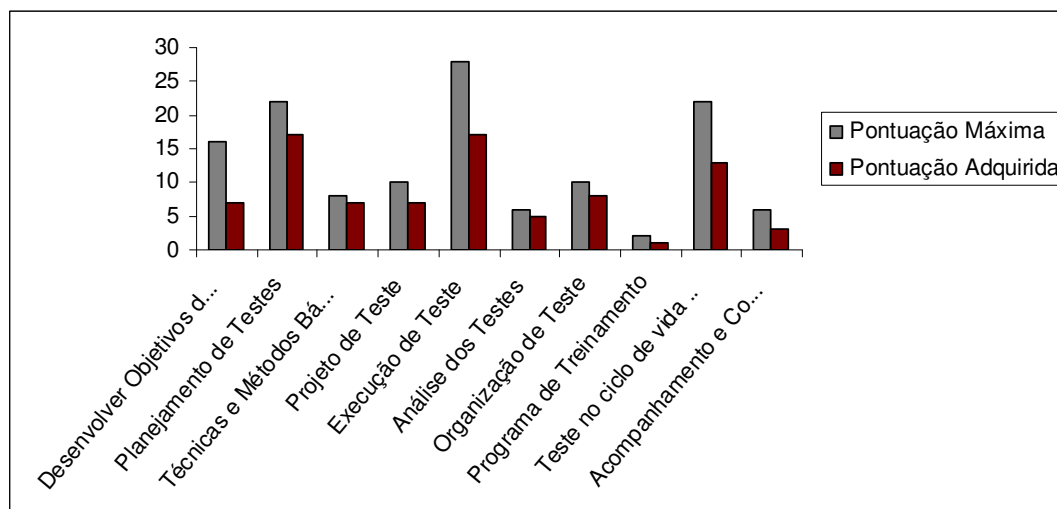
Pela tabela 14, pode-se verificar que as fases Implementar Ações foi a que mais consumiu esforço na aplicação da abordagem por ser execução e o treinamento das ações de melhoria. É importante destacar que em relação a capacidade produtiva da empresa, o custo para implementação e melhoria do processo da empresa foi considerado baixo pela organização.

Ao final do programa de melhoria, o diagnóstico do processo de teste foi realizado novamente com o objetivo de verificar sua evolução com a implementação da abordagem AITS. O resultado final da avaliação pode ser observado pela tabela 15 e 16.

Tabela 15 - Resultado da avaliação após a implementação da AITS – Estudo de Caso 1

RESULTADO DA AVALIAÇÃO		
PONTOS	Nível 2	Nível 3
Total de Pontos Válidos	90	40
Total de Pontos Adquiridos	60	25
RESULTADO	REPROVADO	REPROVADO

Tabela 16 - Áreas de processo após a implementação da AITS – Estudo de Caso 1



Como pode ser observado através das tabelas 15 e 16, o processo não conseguiu atingir o nível 2 do TMM, conseqüentemente o nível 3 também não foi atingido. O nível 2 não foi atingido principalmente porque a área de *debugging* da organização não foi desenvolvida conforme o esperado. Por muitas práticas desta área de processo serem de responsabilidade de outra equipe da organização, foi difícil para a equipe de testes atuar com rigor e eficácia na sua implementação. Além disso, outro empecilho para o alcance do nível 2 ocorreu porque a última iteração que previa, principalmente, a definição de requisitos e casos de testes não-funcionais, execução e implementação de testes não-funcionais e seleção de ferramentas não foi realizada, impactando diretamente na área de processo Execução de Teste. Apesar disso, o processo de teste teve uma melhoria altamente significativa para toda a organização. O processo que antes da implementação

da AITS apresentava 16,67% das práticas TMM nível 2 e 32,5% das práticas TMM nível 3 implementadas, ao final do programa de melhoria apresentava 66% das práticas nível 2 e 62,5% das práticas nível 3 implementadas, deixando o processo de teste da organização muito próximo de alcançar os padrões nível 2 e 3 do TMM. Houve também a percepção de aumento da qualidade do produto 1 por parte da equipe de testes ao final do programa de melhoria. Esta percepção foi coletada através da reunião de fechamento do programa, onde participaram todos os envolvidos com a iniciativa.

6.3 Estudo de Caso 2

O segundo estudo de caso foi realizado numa empresa voltada para o desenvolvimento de projetos para os mais diversos setores. A empresa conta atualmente com aproximadamente 1200 colaboradores e possui a estrutura projetizada.

A abordagem AITS foi aplicada em um projeto desta organização que estava desenvolvendo uma ferramenta de gerência de projetos para uma multinacional do setor de telefonia que seria utilizada por todas as suas filiais da América Latina. A escolha deste projeto para a aplicação da abordagem se deu, principalmente, porque o cliente era estratégico para a organização e se mostrava extremamente insatisfeito com a qualidade dos releases entregues pela equipe. Por causa disso, a alta direção tomou uma série de medidas que entre elas estavam a criação de uma equipe de testes independente para o projeto e a aplicação da abordagem AITS.

Com isso, a equipe do projeto aumentou, contando com a participação de 15 pessoas ao todo, sendo 10 engenheiros de sistemas, 4 engenheiros de teste e um gerente de projeto. Dois membros da equipe de testes estavam trabalhando em outro estado do país, o que, inicialmente, foi um problema para o resto da equipe de teste, já que a comunicação só era possível via MSN ou por telefone, o que muitas vezes dificultava o entendimento das necessidades reais do time e dificultava o relacionamento com o time de desenvolvimento do projeto.

Durante a aplicação da abordagem AITS, a empresa estava se preparando para ser avaliada no CMMI estagiado nível 3. O projeto escolhido não estava dentro do escopo da avaliação, mas mesmo assim deveria seguir uma customização do processo organizacional.

Neste estudo de caso foi utilizada a versão final da abordagem AITS, a fim de validar sua aplicação e comprovar os resultados obtidos. Para iniciar a aplicação da abordagem, o grupo de condutores foi estabelecido, sendo formado por dois membros da equipe de testes do projeto, onde a autora era um deles. Foi definido também que durante os meses do programa de melhoria, os integrantes do time de condutores e teste dedicariam 50% da sua carga horária para as atividades da abordagem.

Na fase de Inicialização, foi obtido o comprometimento entre a alta direção, a gerente do projeto, que representava a alta direção da empresa, e o time independente de testes. Este comprometimento foi realizado através de uma reunião onde foram definidos dois objetivos da organização em relação ao programa de melhoria de testes do projeto:

- Aumentar a qualidade do produto;
- Aumentar a satisfação do cliente.

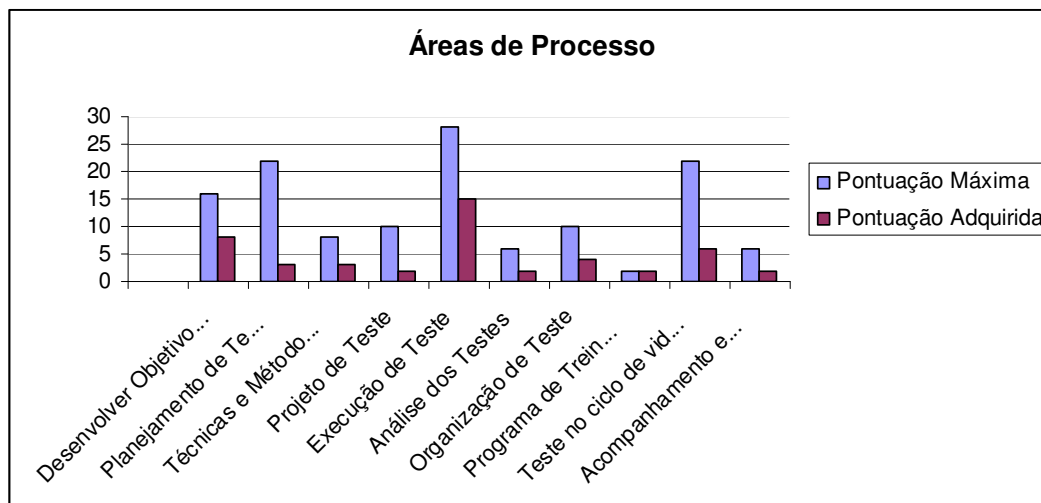
Na fase de Diagnóstico, o questionário de avaliação foi enviado via email para os membros que estavam direta ou indiretamente envolvidos com a área de testes. Os questionários preenchidos foram re-enviados para o time de condutores em um tempo médio de 2 dias. Entre os participantes da avaliação estavam todos os engenheiros de teste do projeto, um analista de requisitos e o gerente de projetos. Os resultados dos checklists de avaliação das entrevistas foram consolidados pelos condutores em um único checklist de avaliação. Na tabela 28 do Apêndice I são apresentadas detalhadamente as informações do checklist de avaliação consolidado por área de processo e nas tabelas 17 e 18 é apresentado resultado final da avaliação.

Resultado Final do Diagnóstico

Tabela 17 - Avaliação do Processo de Teste do Estudo de Caso 2

RESULTADO DA AVALIAÇÃO		
PONTOS	Nível 2	Nível 3
Total de Pontos Válidos	88	38
Total de Pontos Adquiridos	33	14
RESULTADO	REPROVADO	REPROVADO

Tabela 18 - Gráfico do Diagnóstico em relação as áreas de processo do Estudo de Caso 2



Após a consolidação dos dados obtidos, o time de condutores realizou uma reunião, com participação do time do projeto, e a gerente como representante da alta direção da empresa. A apresentação tinha como objetivo exibir os pontos fortes e fracos existentes no processo de testes utilizado pelo projeto. Entre os principais pontos fortes levantados durante a avaliação estavam:

- Existia um processo definido para planejamento, projeto, execução e análise de testes;
- A gerência de configuração do projeto funcionava de maneira adequada;
- Um ambiente adequado para a realização das atividades de testes era fornecido;

- Defeitos eram identificados e registrados no repositório de defeitos;
- Treinamentos organizacionais em testes eram fornecidos.

Entre os principais pontos fracos levantados durante o diagnóstico estavam:

- Embora houvesse um processo de testes definido, ele não era utilizado;
- Testes funcionais e manuais eram realizados de forma exploratória, baseados apenas no conhecimento dos engenheiros de testes, não havia nada documentado.
- Não existia planejamento de testes;
- Testes não-funcionais não eram realizados;
- Testes unitários não eram planejados, projetados e realizados;
- Os requisitos não eram aprovados pelo stakeholders apropriados e os integrantes do time de testes não eram envolvidos apropriadamente;
- Nenhuma métrica de testes estava definida.

Após o fim da fase de Diagnóstico, foi iniciada a fase de Planejamento de Ações de Melhoria. A primeira atividade desta fase foi selecionar as ações a partir do template de ações padrões. As ações selecionadas foram aquelas que endereçavam diretamente as deficiências de teste levantadas durante a avaliação. As ações selecionadas foram priorizadas de acordo com os critérios definidos na abordagem AITS. O resultado da priorização pode ser observado através da tabela 19.

Tabela 19 - Ações Priorizadas

PRIORIZAÇÃO DAS AÇÕES					
	Peso SWOT	Benefício	Custo	Inter dependência	Ordem de Prioridade
Definir template de testes exploratórios	12	4	3	2	21
Realização de Validação de Defeitos	12	4	3	2	21
Especificar testes exploratórios	12	4	3	2	21
Definir template para registro de defeitos	11	3	3	2	19
Definir testes de regressão	11	4	3	0	18
Planejamento dos testes	10	4	2	2	18
Análise do bugs identificados pelo cliente	9	4	3	2	18
Realização de Inspeção dos artefatos de teste	9	4	4	0	17
Planejamento e Acompanhamento diário das atividades	9	4	4	0	17
Definição de papéis e responsabilidades do time de teste	7	3	4	2	16
Inspeção de requisitos pelos membros do time de teste	8	4	2	0	14
Realização de testes de aceitação no ambiente do cliente	8	4	1	0	13
Definição do template de relatórios dos resultados de teste	3	4	4	2	13
Estudo da viabilidade de automação dos testes	7	3	2	0	12
Elaboração de casos de testes	8	3	1	0	12
Definição da ferramenta de gerenciamento de testes	7	3	2	0	12
Realização de relatórios dos Resultados do teste	7	3	2	0	12
Realização dos testes de performance	7	3	1	0	11
Implementar casos de teste de performance	7	3	1	0	11
Definição de métricas básicas de teste	4	3	2	0	9
Definição de template para especificação de CT não funcionais	3	2	3	0	8
Automatizar os casos de teste	2	3	1	0	6

Uma vez realizada a priorização das ações selecionadas, o plano de ação iterativo e incremental foi elaborado considerando as prioridades obtidas para cada ação. No plano de ação foram identificadas quatro iterações para todo o programa de melhoria como é exibido na tabela 20. Este plano considera ainda a realização de pilotos para prática e fixação do conhecimento adquirido durante a iteração. Um plano de ação mais detalhado com responsabilidades, marcos era elaborado no início de cada iteração.

Tabela 20 - Plano de Ação Estudo de Caso 2

Status das Ações		Áreas de Processo				
Ações Finalizadas		DEB_TEST	Objetivos de Debugging e Testes	ANA	Análise de Teste	
Próximas Ações		PLAN	Planejamento de Teste	TEC_MET	Técnicas e Métodos de Teste	
Ações Em Andamento		PROJ	Projetar Teste	CICLO	Ciclo de vida do software	
		TREIN	Treinamento Teste	ACOM	Acompanhamento de Teste	
Ações Planejadas		EXEC	Execução de Teste	ORG	Organização de Teste	
Processo	Ação	Responsável	Observações	Data Inicial	Data Final	Qtd de ReAgend
ITERAÇÃO 1						
PROJ	Definir template para projeto de teste			4/jun	6/jun	
PROJ	Especificação de Testes Exploratórios			7/jun	14/jun	
PROJ	Definir testes para regressão			15/jun	19/jun	
EXEC	Definir template para registro de defeitos			20/jun	21/jun	
EXEC	Validação dos defeitos			22/jun	26/jun	
PILOTO	Piloto: Especificar testes exploratórios, definir testes para regressão e exercitar a validação de defeitos registrados			27/jun	3/jul	
ITERAÇÃO 2						

PLAN	Realização de inspeção dos artefatos de testes			10/jul	12/jul	
PLAN	Planejamento de Testes			13/jul	18/jul	
ANA	Análise dos Bugs identificados pelo cliente			19/jul	20/jul	
ORG	Definir papeis e responsabilidades no time de teste			23/jul	25/jul	
PLAN	planejamento e acompanhamento diário das atividades de teste			25/jul	31/jul	
PILOTO	Piloto: Inspeccionar artefatos produzidos, Analisar bugs do cliente e planejar e acompanhar atividades de teste			31/jul	06/ago	
ITERAÇÃO 3						
PROJ	Definição da ferramenta de gerenciamento de teste			14/ago	17/ago	
PROJ	Estudo da viabilidade de automação de testes			20/ago	22/ago	
CICLO	Inspeção de requisitos por membros da equipe de teste			23/ago	27/ago	
ANA	Definição de template de relatório dos resultados do teste			22/ago	25/ago	
ANA	Realização de relatórios dos resultados do teste			28/ago	31/ago	
PROJ	Especificar Casos de Testes			03/set	14/set	
EXEC	Realização de testes de aceitação no			12/set	3/out	

	ambiente do cliente					
PILOTO	Piloto: Especificar Casos de teste, execução de testes no ambiente do cliente			4/out	12/out	
ITERAÇÃO 4						
ACOM	Definição de métricas básicas para o projeto			05/nov	09/nov	
PROJ	Definição de template para especificação de casos de testes não funcionais			19/nov	22/nov	
EXEC	Implementar casos de teste de performance			22/nov	26/nov	
EXEC	Realização dos testes de performance			27/nov	30/nov	
EXEC	Automatizar os casos de teste			N/A	N/A	
PILOTO	Piloto: Implementação de testes não funcionais, definição e coleta de métricas			10/nov	14/dez	

A fase seguinte, Implementar Ações de Melhorias, foi realizada de forma que toda a equipe de teste estivesse envolvida na execução das ações de melhoria planejadas para a iteração corrente. Ao final de cada iteração, os pilotos previstos foram realizados com sucesso. Devido ao tempo reduzido e à boa qualidade dos artefatos produzidos, os mesmos foram reaproveitados para no projeto real. Para a equipe de testes do projeto, isto foi um ponto positivo, pois ela estava produzindo e utilizando os artefatos, otimizando assim o tempo que eles tinham. Sendo assim, algumas atividades destes pilotos entraram no cronograma do projeto em andamento.

A fase Implementar Ações de Melhoria foi realizada 4 vezes durante o todo o programa de melhoria. A prática de automação dos testes prevista na última iteração

planejada do plano de ação não foi realizada, pois a continuação do projeto foi cancelada o que não justificava mais o esforço que seria gasto para implementação de tal prática. Desta forma, a gerência do projeto juntamente com um representante do time de teste optaram por implementar a prática caso houvesse uma continuação do projeto posteriormente. Entre a iteração 3 e 4 houve um período de aproximadamente 25 dias sem a realização do programa, isto se deve ao fato de que a equipe de testes estava muito sobrecarregada com as atividades do projeto e não podia dedicar 50% de sua carga horária para o programa de melhoria.

Na fase Verificar Resultados e Aprender, reuniões ao final da iteração foram realizadas com a equipe de testes, com a equipe de desenvolvimento, com a engenheira de qualidade e a gerente do projeto. As reuniões tinham duração de aproximadamente quatro horas e as seguintes perguntas eram respondidas por todos os participantes:

- O que deu certo durante a iteração?
- O que pode ser melhor para a próxima iteração?

Todas as respostas eram exibidas e discutidas por todos os presentes na reunião. Ao final da reunião, um dos membros do time de condutores juntamente com a engenheira de qualidade reunia e compilava todas as informações para elas ficassem disponíveis para o time.

Ao final de cada iteração o conjunto de práticas que tiveram aceitação positiva pelo time de teste era agregado ao processo de teste customizado para o projeto, institucionalizando assim a melhoria no âmbito daquele projeto. A fase Acompanhar Melhoria, foi realizada durante todo o programa de melhoria pela gerente do projeto e um membro do time de condutor. Como o gerente do projeto que era também representante da alta direção e estava muito próximo da equipe de condutores e testes, o acompanhamento do andamento do programa era realizado diariamente através de reuniões rápidas de 15 minutos. Com isso, não houve a necessidade de métricas de acompanhamento do programa de melhoria. Na tabela 21 é exibida a duração de cada fase do programa de melhoria.

Tabela 21- Duração de cada fase do programa de melhoria

Fases	Planejado por por Iteração	Realizado por Iteração	Desvio
Inicialização	5 dias	2 dias	-60%
Diagnóstico	2 dias	2 dias	0%
Identificar e Priorizar Melhoria	3 dias	3 dias	0%
Implementar Ações	20 dias	35 dias	75%
Verificar Resultados	2 dias	2 dias	0%
Institucionalizar Melhoria	5 dias	3 dias	-40%
Acompanhamento do Programa	Durante a implementação das outras fases	Durante a implementação das outras fases	Sem desvio

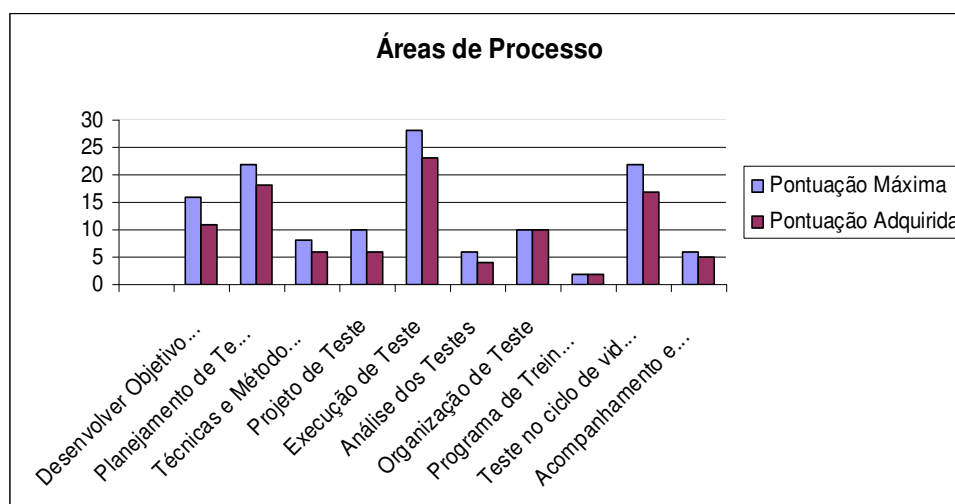
Pela tabela 21, pode-se verificar que a fase Implementar Ações foi a que mais consumiu esforço na aplicação da abordagem. Isto se deve, principalmente, ao fato da equipe de testes ter tido um tempo muito reduzido para realização de atividades do projeto real, o que acabou muitas vezes impactando nas atividades e nas horas do projeto do programa de melhoria, aumentando assim os prazos para finalização das atividades no projeto piloto. Para este estudo de caso, o custo para implementação e melhoria do processo customizado também foi considerado baixo pela organização em relação à capacidade produtiva da empresa e os benefícios trazidos por ela.

Ao final do programa de melhoria, o diagnóstico do processo de teste foi realizado novamente com o objetivo de verificar sua evolução com a implementação da abordagem AITS. O resultado final da avaliação pode ser observado pela tabela 22 e 23.

Tabela 22 - Resultado da avaliação após a implementação da AITS – Estudo de Caso 2

RESULTADO DA AVALIAÇÃO		
PONTOS	Nível 2	Nível 3
Total de Pontos Válidos	84	40
Total de Pontos Adquiridos	68	34
RESULTADO	PASSOU	PASSOU

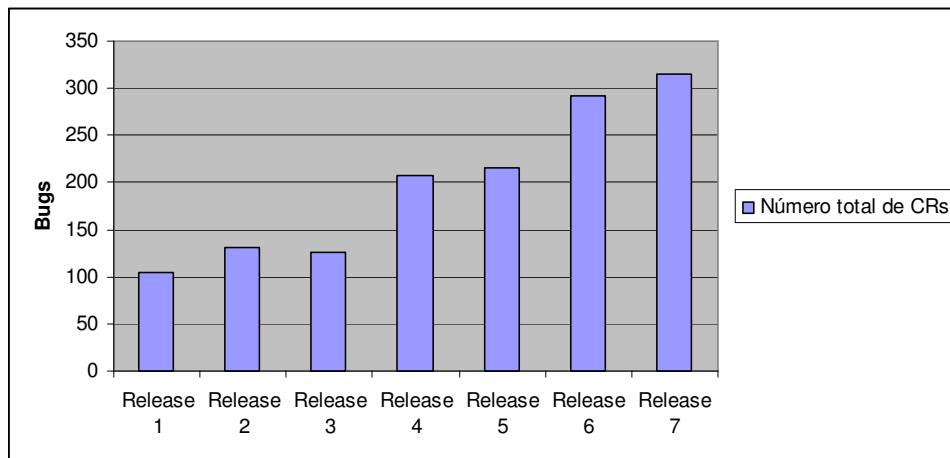
Tabela 23 - Áreas de processo após a implementação da AITS – Estudo de Caso 2



Como pode ser observado através das tabelas 22 e 23, o processo conseguiu atingir o nível 2 e o nível 3 do TMM. Com isso temos ao final do programa de melhoria um processo de testes que atende a 81% do nível 2 e 85% do nível 3 TMM. Algumas melhorias ainda precisarão ser realizadas para que o processo atenda por completo as práticas nível 2 e 3 do TMM, principalmente porque as práticas relacionadas com a automação de testes não foram implementadas, impactando assim na área Execução de Teste.

Além da avaliação do processo em relação ao TMM, um indicador foi estabelecido e coletado ao longo do programa para verificar a eficiência do time de teste. A tabela 20 exibe a quantidade de bugs identificados antes e durante a implementação da abordagem AITS.

Tabela 24 - Número de bugs antes e durante a AITS



Como pode ser observado no gráfico acima, a quantidade de bugs identificados pelo time de teste após a implantação da abordagem AITS (após Release 3) aumentou consideravelmente. O que pode indicar um aumento na qualidade no produto entregue já que mais defeitos foram identificados antes do produto chegar ao cliente. A satisfação do cliente também aumentou consideravelmente ao longo do projeto, se antes da aplicação da abordagem a satisfação do cliente era em média 3,0 ao final da aplicação da abordagem esta satisfação atingiu a média 4,5, sendo que a pontuação máxima possível era 5,0. Além disso, ao final do projeto foi enviado um questionário para os dois representantes do cliente, a fim de analisar suas percepções em relação à qualidade final do produto e como a equipe de testes pode ter influenciado nisto. Os dois representantes do cliente consideraram que a equipe de teste influenciou muito positivamente na qualidade do produto final. Além disso, outro ponto bastante enfatizado por eles foi o envolvimento do time de testes na revisão dos requisitos, ajudando no entendimento das regras de negócio e com isso, realizando testes mais eficazes. As respostas deste questionário podem ser encontradas no Anexo III deste trabalho.

Em relação ao time do projeto, um questionário com mais questões foi enviado para alguns membros da equipe a fim de coletar a percepção deles em relação aos benefícios trazidos pelas mudanças realizadas no time de teste. A grande maioria dos

entrevistados respondeu que as mudanças realizadas ao longo do projeto impactaram diretamente na melhora da qualidade do produto entregue ao cliente. Outro ponto bastante abordado por eles foi o fato de parte da equipe de testes está localizada em outro estado dificultando, inicialmente, a integração entre o time de desenvolvimento e o time de testes. O questionário elaborado juntamente com as respostas dos entrevistados pode ser vistos na íntegra no Anexo III.

6.4 Análise das Práticas Ágeis nos Estudos de Caso

Nesta seção, será analisada a influência de cada prática adotada pela abordagem nos estudos de caso realizados.

Sprint

O conceito de Sprint ajudou no estabelecimento contínuo e iterativo das ações implementadas. Com isso, as mudanças culturais e organizacionais foram inseridas gradativamente atenuando assim a resistência das pessoas envolvidas no programa de melhoria.

Jogo do Planejamento

Cada estudo de caso realizado teve 4 iterações, como explicado anteriormente. Na primeira iteração de cada estudo de caso era realizada a priorização de todas as ações selecionadas para o programa de melhoria. Nas iterações seguintes, era realizada somente uma avaliação para verificar se algum critério de alguma ação priorizada havia sido modificado. Em caso positivo, uma nova priorização era realizada para as ações restantes. Desta forma, esta prática auxiliou na definição de um escopo priorizado para cada Sprint contando com a participação de todo o time de testes e do time de condutores. Com isso, foi possível que todos soubessem da importância e necessidade de cada ação a ser implementada durante a iteração.

Ritmo Sustentável

Esta prática auxiliou na divisão e controle da carga horária dos envolvidos com o programa de melhoria. De forma geral, os programas de melhoria esperam que os envolvidos tenham dedicação full-time no programa. Devido a limitação de recursos presente nas empresas que fazem parte do contexto do trabalho, a abordagem definiu que os envolvidos deveriam dedicar pelo menos 50% da sua carga horária para o programa de melhoria, que não deveria ultrapassar as 8 horas diárias de trabalho para que o andamento do programa não fosse prejudicado. No primeiro estudo de caso, os envolvidos dedicaram fielmente 50% da carga horária para o programa. Já no segundo estudo de caso realizado, na primeira iteração, os envolvidos não dedicaram 50% da sua carga horária devido ao prazo e a condição do projeto. A partir da segunda iteração, isto foi normalizado.

Posse Coletiva e Time Inteiro

Estas práticas ajudaram na integração da equipe com o programa. Elas foram exercitadas principalmente através das reuniões freqüentes e dos pilotos realizados. Todos da equipe de teste, alta direção e equipe de condutores eram envolvidos com o programa de forma efetiva, fazendo com que eles se sentissem parte do programa, podendo opinar, esclarecer dúvidas, sugerir mudanças e etc. Nos dois estudos de caso realizados, esta prática foi absorvida, fazendo com que as equipes respondessem muito bem a implantação destas duas práticas.

Comunicação e Feedback Contínuo

Estas duas práticas foram fundamentais para o bom desempenho do programa de melhoria nos dois estudos de caso realizados. Estas práticas fornecem uma constante e contínua comunicação entre todos do programa, permitindo que os problemas fossem resolvidos rapidamente, permitiu melhor gerenciamento dos riscos e os resultados percebidos em um curto espaço de tempo, pois constantemente feedbacks eram repassados para todos da organização, permitindo a mensuração do sucesso da implantação da abordagem.

Pequenas Versões, Testes de Aceitação e Integração Contínua

Estas práticas possibilitaram que as mudanças fossem inseridas gradativamente através de um pequeno conjunto de ações definidas para a Sprint. Estas pequenas versões implementadas puderam ser testadas, experimentadas através dos projetos pilotos desenvolvidos ao final de cada iteração. As ações que fossem aprovadas pelo time de teste eram integradas ao processo de teste, fazendo com que, ao final da última iteração, o processo contemplasse todas as ações aprovadas durante o programa. Estas práticas foram positivamente utilizadas, possibilitando com que as ações fossem implementadas, testadas e integradas aos poucos e que a equipe obtivesse um melhor entendimento do processo de teste.

Reunião de Retrospectiva da Sprint

Esta prática permitiu que o programa de melhoria fosse continuamente melhorado, já que ao final de cada iteração, todos do time se reuniam para refletir sobre o que tinha sido bom durante a sprint e o que podia ser melhorado para a próxima. Todos compartilhavam opiniões, sugestões e problemas, desta forma as sprints seguintes eram melhoradas evitando os erros cometidos e mantendo os pontos fortes exibidos.

6.5 Considerações Finais

Neste capítulo comprovamos na prática a aplicação da abordagem AITS, pôde-se verificar através dos dois estudos de casos apresentados que é possível realizar uma melhoria de processo de teste a um custo baixo e com efetividade. Além disso, os estudos de casos foram importantes para definir, aplicar e avaliar a abordagem AITS.

Ao final da aplicação da abordagem, a alta direção e a equipe de testes das empresas participantes, consideraram a abordagem como sendo uma boa alternativa para a realização de implantação e/ou melhoria de processo de testes. As principais vantagens listadas por eles em relação a AITS foram:

- Baixo custo de implementação;
- É bastante simples, fácil entendimento;
- Resultados alcançados em um curto espaço de tempo;
- Participação de toda a equipe de testes na implementação de novas práticas;
- Baixa resistência das equipes envolvidas.

Assim, pelos resultados obtidos dos estudos de casos, pode-se dizer que a abordagem AITS se mostra uma boa alternativa para organizações que desejam alcançar melhores resultados na qualidade do produto e na satisfação dos clientes.

Apesar disso, algumas dificuldades foram encontradas ao longo da implantação da abordagem AITS nos estudos de caso relatados, tais como:

- Grande resistência inicial por parte de outras equipes na implantação de uma equipe de testes independente;
- Conflitos entre iniciativas nas empresas, as duas empresas estavam realizando outras iniciativas em melhoria de software, o que, muitas vezes, impedia grandes alterações em seus processos de teste;
- Em um dos projetos, houve reaproveitamento dos artefatos dos projetos pilotos devido à limitação de tempo;
- No estudo de caso 2, houve interrupção do programa de melhoria por sobrecarga da equipe no projeto real.

7

Conclusão

Este capítulo apresenta as considerações finais sobre o trabalho desenvolvido nesta dissertação. A Seção 6.1 apresenta as principais contribuições deste trabalho. Na Seção 6.2, encontra-se as maiores dificuldades encontradas e finalmente a Seção 6.3 apresenta algumas propostas para trabalhos futuros.

7.1 Considerações finais e Principais Contribuições

Este trabalho apresentou a abordagem AITS, cujo objetivo é contribuir para a melhoria de processo de teste de software em organizações de desenvolvimento de software, através da implantação e melhoria de processos de teste baseada em práticas ágeis, considerando suas características e limitações.

Para isso, a abordagem conta com um guia detalhado e provê um suporte efetivo para sua aplicação, incluindo: descrição de cada fase e atividade, papéis envolvidos e templates de documentos.

A abordagem foi baseada em estudos realizados na área de melhoria de processos de teste de software, em outros modelos disponíveis na literatura e na experiência adquirida em implantação de processos de testes em organizações brasileiras.

O grande benefício deste trabalho é prover uma alternativa as empresas que desejam melhorar seus processos e a qualidade dos seus produtos.

Comparando a abordagem AITS com outras existentes, ela apresenta as seguintes vantagens:

- Usa um vocabulário simplificado e evita tanto quanto possível utilizar terminologias técnicas, pois alguns termos utilizados em modelos clássicos se mostram, muitas vezes, de difícil compreensão;
- Os modelos de documentos elaborados são simples o suficiente para evitar que a utilização da abordagem seja complexa e envolva um trabalho longo e burocrático;
- A abordagem se mostrou muito atrativa para as organizações onde a empregamos por ser extremamente simples;
- É específica para o contexto de teste de software, evitando trabalho com grandes adaptações da abordagem;
- Está fortemente ligada às metas de negócio e melhoria da organização em que é aplicada;
- Oferece suporte explícito para sua execução;
- Reúne os benefícios dos modelos de melhoria de processos com os benefícios trazidos pelo modelo de referência em testes, TMM. Fornecendo o “como fazer” e “o que fazer” para obter sucesso em um programa de melhoria de processo de teste;
- Possui investimento compatível com a realidade das empresas;
- Possibilita a visualização de resultados em um curto espaço de tempo.

A tabela 25 apresenta uma tabela comparativa entre os modelos apresentados e a abordagem proposta neste trabalho, seguindo os critérios estabelecidos para a comparação.

Tabela 25- Comparação entre os modelos e a AITS

Crítérios	IDEAL	ISO 15504-4	Pro2Pi	AITs
Custos de acordo com a realidade da empresa				
Ser de fácil compreensão				
Ser flexível				
Baseados em modelos				
Resultados rápidos				
Modelo Incremental				

Através da implantação desta abordagem a melhoria de processos de testes é alcançada de forma gradual, minimizando a resistência das pessoas afetadas, aproveitando os recursos internos da organização, motivando os envolvidos com resultados em curto prazo, evitando abandono do programa de melhoria por falta no entendimento do modelo e evitando adaptações na abordagem para sua aplicação.

A elaboração desta abordagem está fundamentada nas melhores práticas do IDEAL, da ISO/IEC 15504-4, do Pro2Pi e do TMM. Além disso, a abordagem agrega práticas e princípios de metodologias ágeis, como XP e Scrum.

Outro ponto bastante importante apresentado neste trabalho foi do checklist de diagnóstico deixando-o mais objetivo e com o vocabulário mais próximo da realidade das pessoas envolvidas. Além disso, a geração automática do resultado da avaliação baseada nas repostas dos entrevistados, foi um benefício bastante relevante para a

abordagem, já que isso torna a avaliação mais confiável garantindo mais eficácia e maior rapidez no resultado da avaliação.

Os estudos de casos, presente no capítulo 6, demonstraram que aplicação da abordagem AITS é viável no contexto proposto. Assim, acreditamos que por meio da AITS, empresas que se encaixam no contexto descrito neste trabalho possam realizar a melhoria ou implantação do processo de teste com sucesso, através das fases, atividades e templates presentes nesta abordagem.

7.2 Dificuldades Encontradas

Durante o desenvolvimento do trabalho foram encontradas algumas dificuldades, dentre elas estão:

- Necessidade de suporte mais efetivo nas equipes afetadas, pois houve uma maior resistência delas em relação à execução de algumas ações de melhoria;
- Dificuldade de aplicação da abordagem em mais empresas. Devido ao acesso restrito à alta direção das empresas e a pouca cultura de testes de software que as empresas nacionais ainda apresentam, foi difícil conseguir apoio para a aplicação da abordagem AITS em um maior número de empresas de desenvolvimento de software.
- Dificuldade de encontrar literatura sobre Modelo de Maturidade de Testes, apesar de o TMM possuir maior número de literatura disponível, isso não quer dizer que seja abundante. A maioria das publicações encontradas aborda o TMM de forma macro não fornecendo os detalhes que eram necessários para o trabalho. A principal e mais completa fonte de informações utilizada sobre o TMM foi o livro, *Practical Software Testing: A process-oriented approach* [Burnstein, 2003].

7.3 Proposta para trabalhos futuros

Há a possibilidade de alguns trabalhos futuros que poderão ser realizados a partir deste trabalho, dentre eles destacam-se:

- Disponibilização da abordagem AITS numa página da internet, incluindo a descrição completa de todas as atividades, templates de documentos, fórum de discussões e espaço para registro de críticas e sugestões.
- Automatizar os templates apresentados de forma que torne mais rápida e eficiente a utilização da abordagem;
- Estender a abordagem AITS para suportar os níveis 4 e 5 do TMM;
- Realizar mais estudos de casos em diferentes tipos de organizações e sem a participação da autora;

8

Referências Bibliográficas

- [Anacleto, 2004] ANACLETO, A., **Método e Modelo de Avaliação para Melhoria de Processo de Software em Micro e Pequenas Empresas**. Dissertação de Mestrado. Universidade Federal de Santa Catarina. 2004.
- [Basili, 1993] BASILI, V. R., CALDIERA G., ROMBACH H. D., **“The Goal Question Metric Approach”**, University of Maryland, 1993
- [Beck, 1999] BECK, K.; ANDRES, C. **Extreme Programming explained: embrace change**. 1. ed. Upper Saddle River: Addison-Wesley, 1999. 189 p.
- [Beck, 2005] BECK, K.; ANDRES, C. **Extreme Programming explained: embrace change**. 2. ed. Upper Saddle River: Addison-Wesley, 2005. 189 p.
- [Begel, 2007] BEGEL, A.; NAGAPPAN N. **Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study**. International Symposium on Empirical Software Engineering and Measurement, 2007.
- [Brodman, 1994] BRODMAN, J.G.; JOHNSON, D.L. **“What Small Business and Small Organisations say about CMM?”** Proceedings of the 16th International Conference on Software Engineering, Sorrento, Italy, May 1994.
- [Brooks, 1987] BROOKS, F. P. **No silver bullet: essences and accidents of Software Engineering**. IEEE. Computer, v. 20, n. 4, 1987. p.

- 10-19.
- [Burnstein, 2003] BURNSTEIN, I. **Practical software testing : a process-oriented approach**. Springer-Verlag New York, Inc, 2003
- [Carosia, 2003] CAROSIA, J.S. **Levantamento da qualidade do processo de software com foco em pequenas organizações**. Dissertação de Mestrado. Instituto Nacional de Pesquisas Espaciais, 2003.
- [CMM, 2008] <http://www.sei.cmu.edu/cmm/>. Acessado em janeiro 2008.
- [CMMI, 2001] CMMI Product Team, **Capability Maturity Model Integration for Systems and Software Engineering, Continuous Representation**, Software Engineering Institute Technical Report CMU/SEI-2002-TR-001, 2001
- [Cockburn, 2002] COCKBURN, A. **Agile software development**. 1. ed. Boston: Addison-Wesley, 2002. 278 p.
- [Deming, 1986] DEMING, W E. **Out of the Crisis**. MIT Center for Advanced Engineering Study, 1986.
- [Diniz et al, 2007] DINIZ, R.; OCHNER, J. **Implantação de uma Metodologia de Testes em Iterações para um Grupo Independente**. Relato de Experiência, SBQS – Simpósio Brasileiro de Qualidade de Software, 2007.
- [Ericson, 1996] ERICSON, T., SUBOTIC A. **A Test Improvement Model**. Artigo Técnico – EUROSTAR, 1996.
- [Freesz, 2004] FREESZ, E.V., **Melhoria do processo de teste com TMM em uma grande empresa**. Trabalho de Especialização em Melhoria de Processo de Software, Universidade Federal de Lavras, 2004.
- [Ferreira, 2007] FERREIRA, A.I.; CERQUEIRA, R., SANTOS, G., et al. **Retorno de Investimento da Melhoria de Processo de Software na BL Informática**. Relato de Experiência, SBQS – Simpósio de Qualidade de Software, 2007.

- [Gelperin, 1988] GELPERIN, D.; HETZEL, B. **The growth of software testing**. Artigo Técnico - ACM, Volume 31, pp 687-695, 1988.
- [Herbsleb, 1994] HERBSLEB, J.; CARLETON, A, ROZUM J. **Benefits of CMM-Based Software Process Improvement: Initial Results**. Technical Report - CMU/SEI-94-TR-013. 1994.
- [Homyen, 1998] HOMYEN, A. **An assessment model to determine test process maturity**. Ph.D. thesis, Illinois Institute of Tcehnology, Chicago, IL, 1998.
- [IDEAL, 2007] www.sei.cmu.edu/ideal/. Acessado em outubro 2007
- [ISO12207, 2002] Internacional Organization for Standarlization. **ISO/IEC 12207:1995/Amd 1:2002**: Information technology – Software life cycle processes. ISO/IEC International Standard, 2002.
- [ISO15504, 2005] Internacional Organization for Standarlization. **ISO/IEC 15504: Information Technology Process Assessment, Part 1 to Part 5**. ISO/IEC International Standard, 2003-2005.
- [ISO/IEC 15504-4, 2004] Internacional Organization for Standarlization. **ISO/IEC 15504: Information Technology Process Assessment, Part 4 Guidance on Using assessment results**. ISO/IEC International Standard, 2004
- [Karlström, 2004] KARLSTROM, D.; RUNESON P.; NORDÉN S. **A minimal test practice framework for emerging software organizations**. Wiley InterScience Journal, 2004.
- [Kniberg, 2007] KNIBERG, H. **Scrum and XP from Trenches: How we do Scrum**. Enterprise Software Development Series, 2006.

- [Koomen, 1999] KOOMEN, T.; POL, M. **Test Process Improvement: A practical Step by Stepguide to structured testing**, Addison-Wesley, 1999.
- [McFeeley, 1996] McFEELEY, R. **IDEAL: A User's Guide for Software Process Improvement**. Relatório Técnico CMU/SEI-96-HB-001, Software Engineering Institute, 1996.
- [MCT_01, 2001] MINISTÉRIO DA CIÊNCIA E TECNOLOGIA. **Qualidade e Produtividade no Setor de Software Brasileiro: Resultados da Pesquisa 2001**. Disponível em: <http://www.mct.gov.br/Temas/info/Dsi/Quali2001/Public2001.htm> Acesso em: 28 nov. 2007.
- [MDA, 2007] MODEL DRIVEN ARCHITECTURE. Disponível em www.omg.org/mda/. Acesso em: 20 de dezembro de 2007.
- [MPS, 2006] Softex, **Modelo de Referência para Melhoria de Processo de Software Brasileiro**. Guia Geral , 2006.
- [NEXT, 2007] NEXT – Núcleo em Excelência em Testes de Software, <http://www.next.org.br/>. Acessado em 02/04/2007.
- [Niasi, 2003] NIAZI, M.; WILSON D.; ZOWGHI Didar. **A model for the implementation of software process improvement: A pilot study**. Artigo Técnico. Proceedings of the Third International Conference On Quality Software (QSIC'03). IEEE Software, 2003.
- [Ochner, 2007] OCHNER, J. **Guia para Implantação de Testes em Pequenas e Médias Empresas de Software**. Artigo Técnico, EBTS – Evento Brasileiro de Testes de Software, 2007.
- [Potts, 1993] POTTS, C. Software-Engineering Research Revised. IEEE Software, Volume 10, Number 5, pages 19-28, September 1993.
- [Poppendieck, 2003] POPPENDIECK, M.; POPPENDIECK, T. **Lean software**

- development: an agile toolkit.** 1. ed. Upper Saddle River, NJ: Addison-Wesley, 2003
- [Reffson, 2006] REFFSON A.; BEZERRA C.I.; COUTINHO E. **Análise da Aderência de um Processo de Teste ao TMM.** Artigo Técnico, SBTS – I Simpósio Brasileiro de Teste de Software, 2006.
- [Rico, 2002] RICO, D. F. **Software process improvement: Modeling return on investment (ROI).** Software Engineering Institute (SEI) Software Engineering Process Group Conference (SEPG 2002), Phoenix, Arizona, 2002.
- [Rising, 2000] RISING, L.; JANOFF, N.S. **Development Process for Small Team.** IEEE Software, Agosto 2000.
- [Salviano, 2006] SALVIANO, C. F.; **Uma Proposta orientada a Perfis de Capacidade de Processo para Evolução da Melhoria de Processo de Software.** Tese de Doutorado, Universidade Estadual de Campinas, 2006.
- [Santana, 2007] SANTANA, A.F., **Problemas em Iniciativas de Melhoria de Processos de Software sob a Ótica de uma Teoria de Intervenção.** Dissertação de Mestrado, Universidade Federal de Pernambuco, 2007.
- [Santos, 2006] SANTOS J.; ALVES G. **TMM-e Framework.** Artigo Técnico, SBTS – I Simpósio Brasileiro de Teste de Software, 2006.
- [Schwaber, 2004] SCHWABER, K. **Agile Project Management with Scrum.** 1 ed. Microsoft Press, WA, 2004.
- [Setúbal, 2007] SETÚBAL, C. F.; CORTÉS, M. I.; SANTOS, V. A. **Estabelecendo a Cultura de Utilização de Processos de Software em uma Micro-Empresa.** Artigo Técnico, WOSES - III Workshop Um Olhar Sociotécnico sobre a Engenharia de Software, 2007.

- [Ste_01] **ImproveIt**, <http://www.improveit.com.br>
Acessado em setembro de 2007
- [SOFTEX, 2007] SOFTEX – Sociedade Brasileira para Promoção e Exportação de Software, <http://recife.softex.br/>. Acessado em outubro de 2007.
- [Ste_03, 2007] MANIFESTO ÁGIL. <http://agilemanifesto.org/>. Acessado em setembro 2007
- [Ste_06, 2008] http://www.pronus.eng.br/artigos_tutoriais/processo_desenvolvimento/melhoria_processo.php?pagNum=0#ref_mcconnell. Acessado em janeiro de 2008.
- [Sommerville, 2006] SOMMERVILLE, I. **Software Engineering**. 8 ed. Addison Wesley, WA, 2006.
- [Teles, 2004] TELES, V. M. **Extreme Programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade**. 1. ed. São Paulo: Novatec, 2004.
- [Teles, 2005] TELES, V. M. **Um Estudo de Caso da Adoção das Práticas Ágeis e Valores do Extreme Programming**. Dissertação de Mestrado. Universidade Federal do Rio de Janeiro. 2005
- [Vasconcelos, 2007] VASCONCELOS, A. **Extreme Programming**. Disponível em : <http://www.cin.ufpe.br/~if719/slides/XP.PDF>. Acesso em: 30 de dezembro de 2007.
- [Vasconcelos et al, 2006] VASCONCELOS, A. M.; ROUILLER, A.C. et al. **Introdução à Engenharia de Software e à Qualidade de Software**. Lavras: UFLA/FAEPE, 2006.
- [Veenendaal, 2007] VEENENDAAL, E.; GROOF, R.; HENDRIKS R. **Test Process Improvement using TMM**. Artigo Técnico,

- STAR Tester, 2007.
- [XP, 2007] Extreme Programming – Programação Extrema, <http://www.extremeprogramming.org/>. Acessado em setembro de 2007
- [Weber, 2005] WEBER, S.; VON WANGENHEIM, C.G. **Documentação da abordagem ASPE/MSD e seus Templates**. Relatório Técnico. Laboratório de Qualidade e Produtividade de Software (LQPS), 2005.

Anexo I – Documentos dos Estudos de Caso e Auxiliares

Tabela 26 - Mapeamento entre as Áreas de processo do TMM e as questões do checklist do AITS

Área de Processo do TMM	Subprática do TMM	Questão AITS
Desenvolver Objetivos de Teste e Debugging	SP1: um grupo de teste e debugging é estabelecido. Políticas e objetivos são definidos	1- Existe política organizacional estabelecida para planejar e realizar teste?
		2- Testes são realizados pela equipe de desenvolvimento antes de serem liberados para equipe de teste?
		3- Pápeis e responsabilidades para as atividades relacionadas a testes unitários estão estabelecidos?
		4- Existe uma equipe independente totalmente dedicada às atividades de teste na organização?
	SP2: Teste e debugging são refletidos no plano de teste	1- O plano de testes possui conteúdo apropriado? - Estágios e Tipos de Teste
	SP3: Classificação de defeitos e um repositório de defeitos são estabelecidos	1- Defeitos são identificados e registrados apropriadamente por testadores e usuários? . São registrados: Defeito, Passos para reprodução . A análise e gestão dos defeitos identificados é realizada?
		2- Registro e análise de Impacto, Severidade, Prioridade, Ações de correção, Responsáveis são realizadas?
		3- Ferramentas apropriadas são utilizadas para apoiar a gestão dos defeitos identificados?
	SP4: Métricas básicas são identificadas e coletadas	1- Mettricas básicas são identificadas, coletadas e analisadas ?

Tabela 21A – Continuação do Mapeamento entre as Áreas de processo do TMM e as questões do checklist do AITS

Área de Processo do TMM	Subprática do TMM	Questão AITS
Planejamento de Testes	SP6: Um grupo para planejamento de teste é estabelecido. Políticas e Objetivos são definidos para planejamento de testes	1- Existe uma equipe independente totalmente dedicada às atividades de teste na organização?
		2- Recursos apropriados (esforço, prazo, ferramentas, ...) são disponibilizados para o planejamento de testes?
		3- O planejamento dos testes é realizado em momento apropriado do ciclo de vida ?
		4- Existe política organizacional estabelecida para planejar e realizar teste?
	SP7: Planos de teste para todos os níveis de teste são elaborados, armazenados e distribuídos. Outros documentos de teste são identificados e elaborados.	1- O plano de testes possui conteúdo apropriado? . Informações existentes no plano: - Estratégia de Teste - Cobertura dos testes - Estágios e Tipos de testes
		2- O planejamento dos testes é documentado e está coerente com o Plano do Projeto?
		3- Os artefatos de planejamento de testes são gerenciados e controlados?
		4- Os casos de testes para requisitos funcionais e não funcionais são identificados e documentados?
		5- Os procedimentos de testes são identificados documentados?
		6- Os resultados dos testes são registrados em um relatório resumo?
		7- Um relatório de release é elaborado e entregue juntamente com o produto?
	SP8: Treinamento é disponibilizado para ensinar a utilização e a elaboração do plano de teste	1- Treinamento necessário é provido para os profissionais envolvidos com as atividades de teste?
	SP9: Requisitos são considerados como entrada do plano de teste	1- O plano de testes possui conteúdo apropriado? . Informações existentes no plano: - Requisitos a serem testados (iterações) - Priorização dos requisitos - Requisitos não testados
	SP10: Ferramentas de planejamento de Testes e métricas básicas são estabelecidas	1- Recursos apropriados (esforço, prazo, ferramentas, ...) são disponibilizados para o planejamento de testes?
		2- Ferramentas para apoiar a o gerenciamento dos testes são utilizadas?
		3- Mettricas básicas são identificadas, coletadas e analisadas ?

Tabela 21B – Continuação do Mapeamento entre as Áreas de processo do TMM e as questões do checklist do AITS

Área de Processo do TMM	Subprática do TMM	Questão AITS
Técnicas e Métodos Básicos para Teste	SP11: Treinamentos e ferramentas são disponibilizadas para dar suporte métodos e técnicas de teste	1- Treinamento necessário é provido para os profissionais envolvidos com as atividades de teste?
	SP12: Testes unitários, integração, sistema e aceitação são planejados e implementados	1- Os estágios de teste unitário, de integração, de sistema, aceitação são planejados, documentados e realizados?
	SP13: Estratégias básicas de testes, técnicas e métodos são utilizadas para projetar testes	1- Técnica/estratégia apropriada é utilizada para identificar/documentar casos, procedimentos e massa de testes?
		2- Os procedimentos de testes são identificados e documentados?
Organização de Teste	SP14: Um grupo de teste é estabelecido. Liderança e suporte é fornecido para o time de testes. Papéis, responsabilidades e carreiras são definidas para o time de teste	1- Existe uma equipe independente totalmente dedicada às atividades de teste na organização?
		2- Papéis e responsabilidades relacionados ao processo de testes e qualidade do produto estão estabelecidos?
		3- Existe um plano de cargos e salários para os componentes desta equipe?
	SP15: O grupo de teste é estabelecido e identificado na hierarquia da organização. Pessoas treinadas e motivadas são adicionadas ao grupo de teste. Canais de comunicação com outros grupos são estabelecidos	1- A equipe independente de teste está indentificada na estrutura organizacional da organização como um todo?
		2- As mudanças aos requisitos são avaliadas e comunicadas apropriadamente junto à equipe de testes?
		3- O controle de versão e mudanças do produto entregue à equipe de testes é realizado apropriadamente?
		4- O comprometimento com o plano de testes é obtido dos stakeholders relevantes?
	SP16: Treinamento é fornecido para o time de teste.	1- Treinamento necessário é provido para os profissionais envolvidos com as atividades de teste?

Tabela 21C – Continuação do Mapeamento entre as Áreas de processo do TMM e as questões do checklist do AITS

Área de Processo do TMM	Subprática do TMM	Questão AITS
Programa de Treinamento	SP17: Um grupo de treinamento é estabelecido. Este grupo elabora e distribui uma política de treinamento.	1- Existe uma política organizacional estabelecida para os testes?
		2- Um ambiente adequado às necessidades do testes a serem realizados (laboratórios, hardware, software, ferramentas, treinamentos) é identificado e viabilizado?
	SP18: Um programa de treinamento é desenvolvido.	1- Treinamento necessário é provido para os profissionais envolvidos com as atividades de teste?
Teste no ciclo de vida do software	SP19: Um grupo é estabelecido para integrar testes no ciclo de vida do software. Políticas, procedimentos são elaborados.	1- Existe uma política organizacional estabelecida para os testes?
		2- Papéis e responsabilidades relacionados ao processo de testes e qualidade do produto estão estabelecidos?
	SP20: Atividades de testes são integradas ao ciclo de vida do software utilizando um modelo de ciclo de vida?	1- O processo definido para testes utiliza modelo/metodologia apropriada como base?
	SP21: Recursos e treinamentos são fornecidos para suportar as atividades de teste integradas.	1- O plano de testes possui conteúdo apropriado? - Recursos (Equipe, Ferramentas e Ambientes)
		2- Treinamento necessário é provido para os profissionais envolvidos com as atividades de teste?
Acompanhamento e Controle do Processo de Teste	SP22: Um grupo para controlar e monitorar teste é formado. O grupo elabora políticas e métricas para controle e monitoramento dos testes	1- O processo de teste e a qualidade do produto são monitorados e métricas básicas são consideradas para melhoria contínua?
		2- A conformidade das atividades de teste são verificadas com relação à conformidade com procedimentos, padrões e critérios estabelecidos?
		3- Papéis e responsabilidades relacionados ao processo de testes e qualidade do produto estão estabelecidos?
	SP23: Métricas para controlar e monitorar teste são coletadas. Planos de contingência são elaborados quando desvios são encontrados	1- Mettricas básicas são identificadas, coletadas e analisadas ?
		2- O plano de testes possui conteúdo apropriado? . Informações existentes no plano: - Riscos, Mitigação e Contingência
	SP24: Ferramentas e recursos são disponibilizados para controlar e monitorar os testes	1- O plano de testes possui conteúdo apropriado? - Recursos (Equipe, Ferramentas e Ambientes)

Tabela 27 - Avaliação do Processo de Teste do Estudo de Caso 1

Área de Processo	Questão	Grau de Definição
Desenvolver Objetivos de Teste e Debugging	Existe política organizacional estabelecida para testes?	Não
	Existe um processo definido e institucionalizado para planejamento, projeto, execução e análise de teste?	Parcialmente
	Testes unitários são planejados?	Não
	Papéis e responsabilidades para as atividades relacionadas a testes unitários estão estabelecidos	Parcialmente
	Testes unitários são projetados?	Parcialmente
	Testes unitários são executados?	Sim
	O resultado dos testes unitários são registrados/avaliados e problemas são corrigidos?	Não
	Os artefatos de testes unitários são gerenciados e controlados	Não
Planejamento de Testes	o planejamento dos testes é documentado e está coerente com o Plano do Projeto?	Não
	Testes são realizados pela equipe de desenvolvimento antes de ser liberados para a equipe de testes? Quais tipos e estágios?	Parcialmente
	O plano de testes possui conteúdo apropriado? . Informações existentes no plano: - Estratégia de Teste - Cobertura dos testes - Estágios e Tipos de testes - Produtos de trabalho a serem testados (casos de uso, classes, métodos etc) - Requisitos a serem testados (iterações) - Priorização dos requisitos - Recursos (Equipe, Ferramentas e Ambientes) - Requisitos não testados - Responsáveis pelos testes - Riscos, Mitigação e Contingência - Critérios de aceitação	Não
	Recursos apropriados (esforço, prazo, ferramentas, ...) são disponibilizados para o planejamento de testes?	Não
	O planejamento dos testes é realizado em momento apropriado do ciclo de vida (antes da execução dos testes) e mantido atualizado ao longo do projeto? --> Analisar se o esforço e equipe estimados é apropriado	Não
	Os estágios de teste unitário, de integração, de sistema, aceitação são planejados, documentados e realizados?	Parcialmente
	Os tipos (funcional, performance, stress, segurança, configuração, ...) de testes apropriados são planejados, documentados e realizados?	Não
	As abordagens caixa-branca e caixa-preta são utilizadas?	Parcialmente

	O plano de testes é inspecionado/revisado por stakeholders apropriados?	Não
	O comprometimento com o plano de testes é obtido dos stakeholders relevantes? --> Comunicação e comprometimento de todos os envolvidos	Não
	Os artefatos de planejamento de testes são gerenciados e controlados?	Não
Técnicas e Métodos Básicos para Teste	O processo definido para testes utiliza modelo/metodologia apropriada como base?	Não
	Técnica/estratégia apropriada é utilizada para identificar/documentar casos, procedimentos e massa de testes?	Não
	Ferramentas apropriadas para automação de testes são utilizadas?	Não
	Ferramentas para apoiar a o gerenciamento dos testes é utilizada? --> Inclui: registro de Casos de Teste, procedimentos e massa de dados, registro de suites, resultados, controle de acesso, ...	Não
Projeto de Teste	Os casos de testes para requisitos funcionais e não funcionais são identificados documentados?	Parcialmente
	Os procedimentos de testes são identificados documentados?	Parcialmente
	A massa de dados de testes é identificada/documentada?	Não
	Os casos, procedimentos e massa de dados de testes são gerenciados, controlados e atualizados de acordo com a atualização dos requisitos?	Não
	Casos de Testes são automatizados apropriadamente? --> Quais os tipos de testes automatizados? --> Qual o critério utilizado para selecionar casos de testes a serem automatizados?	Não
Execução de Teste	Um ambiente adequado às necessidades do testes a serem realizados (laboratórios, hardware, software, ferramentas, treinamentos) é identificado e viabilizado?	Parcialmente
	Smoke tests são identificados? --> Testes para validar a Build de testes	Não
	A Suite de Testes a ser realizada é planejada antes de cada execução? Como este planejamento é documentado?	Não
	Testes manuais são executados e resultados são registrados?	Parcialmente
	Testes automáticos são executados e resultados são registrados?	Não
	Testes funcionais são realizados?	Parcialmente
	Testes de Carga/Volume são realizados? --> Avalia performance do software sobre condições normais de uso. Exemplo: número de transação por unidade de tempo, tempo de resposta, usuários simultâneos, .. --> Como o sistema trabalha com grande volume de dados, transação, usuários, periféricos	Não

	Testes de Stress são realizados? --> Testa condições anormais. Inclui: cargas alta de dados, usuários, memória insuficiente, recursos não disponíveis (impressora, por exemplo), ...	Não
	Testes de Configuração/Instalação são realizados? --> Se o software funciona no hardware e software proposto --> Se o software instala conforme planejado em diferentes hardware/software e sobre condições diferentes (pouco espaço, interrupção, ...)	Não
	Testes de Segurança são realizados? --> Se dados e sistemas não acessados apenas pelos possíveis atores	Não
	Defeitos são identificados e registrados apropriadamente por testadores e usuários? . São registrados: Defeito, Passos para reprodução	Não
	Defeitos identificados apenas pelos usuários são analisados criticamente quanto à realização dos testes internos --> Planos, estratégias, casos e procedimentos de testes são revistos --> Identificar motivos pelos quais estes problemas não são identificados pela equipe	Não
	A análise e gestão dos defeitos identificados é realizada? . Registro e análise de Impacto, Severidade, Prioridade, Ações de correção, Responsáveis	Não
	Ferramentas apropriadas são utilizadas para apoiar a gestão dos defeitos identificados?	Não
Análise dos Testes	Os resultados dos testes são analisados e comparados aos resultados esperados e critério de aceitação?	Não
	Os resultados dos testes são registrados em um relatório resumo e apresentado aos stakeholders relevantes. --> Informações típicas do relatório: . Casos de testes realizados . Métricas de testes . Bugs encontrados . Desvios ao planejamento	Parcialmente
	Um relatório de release é elaborado e entregue juntamente com o produto? --> O documento descreve bugs encontrados, versão do produto, restrições de ambiente.	Parcialmente
Organização de Teste	Existe uma equipe independente totalmente dedicada às atividades de teste na organização?	Sim
	A equipe independente de teste está identificada na estrutura organizacional da organização como um todo?	Parcialmente
	Papéis e responsabilidades relacionados ao processo de testes e qualidade do produto estão estabelecidas?	Parcialmente

	Existe um plano de cargos e salários para componentes desta equipe?	Parcialmente
	As atividades, situação atual e resultados do processo de teste são acompanhados com a gerência sênior e questões são resolvidas?	Não
Programa de Treinamento	Treinamento necessário é provido para os profissionais envolvidos com as atividades de teste?	Parcialmente
Teste no ciclo de vida do software	Papéis e responsabilidades para identificação e documentação dos requisitos estão estabelecidas?	Parcialmente
	Os requisitos são identificados/documentados?	Sim
	Técnicas apropriadas são utilizadas para levantar e especificar os requisitos?	Parcialmente
	Os requisitos funcionais e não funcionais são especificados em um grau de detalhe suficiente?	Não
	Os requisitos são revisados/validados sobre critérios de aceitação dos requisitos estabelecidos pelos stakeholders apropriados?	Parcialmente
	Os responsáveis pelos testes se envolvem apropriadamente com os requisitos?	Não
	Há rastreabilidade definida entre os requisitos e os elementos do produto?	Não
	Mudanças aos requisitos são gerenciadas? . Solicitações de mudanças são registradas, avaliadas, impacto é registrado, documentos impactados atualizados?	Parcialmente
	As mudanças aos requisitos são avaliadas e comunicadas apropriadamente com a equipe de testes?	Não
	Ferramenta apropriadas são utilizadas para documentar/gerenciar os requisitos?	Não
	O controle de versão e mudanças do produto entregue aos testes é realizado apropriadamente?	Parcialmente
	Métricas básicas são identificadas, coletadas e analisadas ?	Não
Acompanhamento e Controle do Processo de Teste	O processo de teste e a qualidade do produto são monitorados e métricas básicas são consideradas para melhoria contínua?	Não
	A conformidade das atividades de teste são verificadas com relação à conformidade com procedimentos, padrões e critérios estabelecidos?	Não

Tabela 28 - Avaliação do Processo de Teste do Estudo de Caso 2

Área de Processo	Questão	Grau de Definição
Desenvolver Objetivos de Teste e Debugging	Existe política organizacional estabelecida para testes?	Sim
	Existe um processo definido e institucionalizado para planejamento, projeto, execução e análise de teste?	Sim
	Testes unitários são planejados?	Não
	Papéis e responsabilidades para as atividades relacionadas a testes unitários estão estabelecidas	Não
	Testes unitários são projetados?	Parcialmente
	Testes unitários são executados?	Parcialmente
	O resultado dos testes unitários são registrados/avaliados e problemas são corrigidos?	Não
	Os artefatos de testes unitários são gerenciados e controlados	Sim
Planejamento de Testes	o planejamento dos testes é documentado e está coerente com o Plano do Projeto?	Não
	Testes são realizados pela equipe de desenvolvimento antes de ser liberados para testes? Quais tipos e estágios?	Não
	O plano de testes possui conteúdo apropriado? . Informações existentes no plano: - Estratégia de Teste - Cobertura dos testes - Estágios e Tipos de testes - Produtos de trabalho a serem testados (casos de uso, classes, métodos etc) - Requisitos a serem testados (iterações) - Priorização dos requisitos - Recursos (Equipe, Ferramentas e Ambientes) - Requisitos não testados - Responsáveis pelos testes - Riscos, Mitigação e Contingência - Critérios de aceitação	Não
	Recursos apropriados (esforço, prazo, ferramentas, ...) são disponibilizados para o planejamento de testes?	Não
	O planejamento dos testes é realizado em momento apropriado do ciclo de vida (antes da execução dos testes) e mantido atualizado ao longo do projeto? --> Analisar se o esforço e equipe estimados é apropriado	Não
	Os estágios de teste unitário, de integração, de sistema, aceitação são planejados, documentados e realizados?	Não
	Os tipos (funcional, performance, stress, segurança, configuração, ...) de testes apropriados são planejados, documentados e realizados?	Não
	As abordagens caixa-branca e caixa-preta são utilizadas?	Parcialmente

	O plano de testes é inspecionado/revisado por stakeholders apropriados?	Não
	O comprometimento com o plano de testes é obtido dos stakeholders relevantes? --> Comunicação e comprometimento de todos os envolvidos	Não
	Os artefatos de planejamento de testes são gerenciados e controlados?	Sim
Técnicas e Métodos Básicos para Teste	O processo definido para testes utiliza modelo/metodologia apropriada como base?	Sim
	Técnica/estratégia apropriada é utilizada para identificar/documentar casos, procedimentos e massa de testes?	Parcialmente
	Ferramentas apropriadas para automação de testes são utilizadas?	Não
	Ferramentas para apoiar a o gerenciamento dos testes é utilizada? --> Inclui: registro de CTs, procedimentos e massa de dados, registro de suites, resultados, controle de acesso, ...	Não
Projeto de Teste	Os casos de testes para requisitos funcionais e não funcionais são identificados documentados?	Não
	Os procedimentos de testes são identificados documentados?	Parcialmente
	A massa de dados de testes é identificada/documentada?	Não se aplica
	Os casos, procedimentos e massa de dados de testes são gerenciados e controladas e atualizados de acordo com a atualização dos requisitos?	Parcialmente
	Casos de Testes são automatizados apropriadamente? --> Quais os tipos de testes automatizados? --> Qual o critério utilizado para selecionar casos de testes a serem automatizados?	Não
Execução de Teste	Um ambiente adequado às necessidades do testes a serem realizados (laboratórios, hardware, software, ferramentas, treinamentos) é identificado e viabilizado?	Sim
	Smoke tests são identificados? --> Testes para validar a Build de testes	Não
	A Suite de Testes a ser realizada é planejada antes de cada execução? Como este planejamento é documentado?	Parcialmente
	Testes manuais são executados e resultados registrados?	Sim
	Testes automáticos são executados e resultados registrados?	Não
	Testes funcionais são realizados?	Sim
	Testes de Carga/Volume são realizados? --> Avalia performance do software sobre condições normais de uso. Exemplo: número de transação por unidade de tempo, tempo de resposta, usuários simultâneos, .. --> Como o sistema trabalha com grande volume de dados, transação, usuários, periféricos	Não

	Testes de Stress são realizados? --> Testa condições anormais. Inclui: cargas alta de dados, usuários, memória insuficiente, recursos não disponíveis (impressora, por exemplo), ...	Não
	Testes de Configuração/Instalação são realizados? --> Se o software funciona no hardware e software proposto --> Se o software instala conforme planejado em diferentes hardware/software e sobre condições diferentes (pouco espaço, interrupção, ...)	Não
	Testes de Segurança são realizados? --> Se dados e sistemas não acessados apenas pelos possíveis atores	Parcialmente
	Defeitos são identificados e registrados apropriadamente por testadores e usuários? . São registrados: Defeito, Passos para reprodução	Sim
	Defeitos identificados apenas pelos usuários são analisados criticamente quanto à realização dos testes internos --> Planos, estratégias, casos e procedimentos de testes são revistos --> Identificar motivos pelos quais estes problemas não são identificados pela equipe	Parcialmente
	A análise e gestão dos defeitos identificados é realizada? . Registro e análise de Impacto, Severidade, Prioridade, Ações de correção, Responsáveis	Sim
	Ferramentas apropriadas são utilizadas para apoiar a gestão dos defeitos identificados?	Sim
Análise dos Testes	Os resultados dos testes são analisados e comparados aos resultados esperados e critério de aceitação?	Parcialmente
	Os resultados dos testes são registrados em um relatório resumo e apresentado aos stakeholders relevantes. --> Informações típicas do relatório: . Casos de testes realizados . Métricas de testes . Bugs encontrados . Desvios ao planejamento	Parcialmente
	Um relatório de release é elaborado e entregue juntamente com o produto? --> O documento descreve bugs encontrados, versão do produto, restrições de ambiente.	Não
Organização de Teste	Existe uma equipe independente totalmente dedicada às atividades de teste na organização?	Parcialmente
	A equipe independente de teste está indentificada na estrutura organizacional da organização como um todo?	Não
	Papéis e responsabilidades relacionados ao processo de testes e qualidade do produto estão estabelecidas?	Parcialmente

	Existe um plano de cargos e salários para componentes desta equipe?	Sim
	As atividades, situação atual e resultados do processo de teste são acompanhados com a gerência sênior e questões são resolvidas?	Não
Programa de Treinamento	Treinamento necessário é provido para os profissionais envolvidos com as atividades de teste?	Sim
Teste no ciclo de vida do software	Papéis e responsabilidades para identificação e documentação dos requisitos estão estabelecidas?	Sim
	Os requisitos são identificados/documentados?	Sim
	Técnicas apropriadas são utilizadas para levantar e especificar os requisitos?	Não
	Os requisitos funcionais e não funcionais são especificados em um grau de detalhe suficiente?	Não
	Os requisitos são revisados/validados sobre critérios de aceitação dos requisitos estabelecidos pelos stakeholders apropriados?	Não
	Os responsáveis pelos testes se envolvem apropriadamente com os requisitos?	Não
	Há rastreabilidade definida entre os requisitos e os elementos do produto?	Não sei
	Mudanças aos requisitos são gerenciadas? . Solicitações de mudanças são registradas, avaliadas, impacto é registrado, documentos impactados atualizados?	Não
	As mudanças aos requisitos são avaliadas e comunicadas apropriadamente com a equipe de testes?	Não
	Ferramentas apropriadas são utilizadas para documentar/gerenciar os requisitos?	Não
Acompanhamento e Controle do Processo de Teste	O controle de versão e mudanças do produto entregue aos testes é realizado apropriadamente?	Sim
	Métricas básicas são identificadas, coletadas e analisadas ?	Não
	O processo de teste e a qualidade do produto são monitorados e métricas básicas são consideradas para melhoria contínua?	Não
	A conformidade das atividades de teste são verificadas com relação à conformidade com procedimentos, padrões e critérios estabelecidos?	Sim

Anexo II – Templates

Ata de Reunião

Redator: <nome do redator da ata>	Data/Hora Início: <hora início>
Local: <local onde foi realizada a reunião>	Data/Hora Fim: <hora fim>

1. OBJETIVO

<objetivo da reunião>

2. PARTICIPANTES

<nomes e papéis dos participantes da reunião>

3. TÓPICOS DISCUTIDOS

<assunto discutido durante a reunião>

4. PRÓXIMAS AÇÕES

<detalhar as ações futuras e os responsáveis por cada uma delas>

5. PRÓXIMA REUNIAO

<data da próxima reunião, se for o caso>

6. OBSERVAÇÃO

<detalhar observações, se for o caso>

7. ASSINATURAS

<assinatura das pessoas presentes na reunião>

Esta ata foi redigida por <Nome Redator/e-mail/fone>. Qualquer dúvida ou discordância, favor entrar em contato com o redator.

Itens de uma apresentação de Kickoff

- **Slide 1** – Conteúdo da Apresentação;
- **Slide 2** – Visão Geral do Programa, incluindo Objetivos, Metas e Benefícios;
- **Slide 3** – Escopo do Programa;
- **Slide 4** – Escopo Negativo;
- **Slide 5** – Fatores Críticos de Sucesso, incluindo premissas para a realização do programa;
- **Slide 6** – Equipe do Programa, incluindo as equipes afetadas;
- **Slide 7** – Riscos.

Relatório Final de Avaliação

	1. DATA DA AVALIAÇÃO <Data da realização das entrevistas de avaliação> 2. LOCAL DA AVALIAÇÃO <Local onde foi realizada a avaliação> 3. EQUIPE DE CONDUTORES <Nomes dos condutores responsáveis pela avaliação> 4. ENTREVISTADOS <Nomes e papéis dos entrevistados da avaliação> 5. TIPO DE AVALIAÇÃO <Indicar como foi realizada a avaliação, através de entrevistas presenciais ou preenchimento do checklist via e-mail > 6. RESUMO DOS PONTOS FORTES DO PROCESSO AVALIADO <Indicar e descrever os principais pontos fortes do processo avaliado > 7. RESUMO DAS OPORTUNIDADES DE MELHORIA DO PROCESSO AVALIADO <Indicar e descrever as principais oportunidades de melhoria do processo avaliado> 8. RESULTADO OBTIDO PELO PROCESSO AVALIADO <Colocar o gráfico que resume o resultado da avaliação do processo>
--	---

Itens da Apresentação de Avaliação

- **Slide 1** – Conteúdo da Apresentação;
- **Slide 2** – Objetivos da Avaliação;
- **Slide 3** – Escopo da Avaliação;
- **Slide 4** – Entendimento do Diagnóstico, incluindo definição do que é um diagnóstico e como foi realizado;
- **Slide 5** – Equipe Envolvida, incluindo os entrevistadores e os entrevistados, detalhando cargos e responsabilidades;
- **Slide 6** – Informações Gerais sobre a Avaliação;
- **Slide 7** – Pontos Fortes diagnosticados do processo avaliado;
- **Slide 8** – Pontos de Melhoria diagnosticados do processo avaliado;
- **Slide 9** – Resultado obtido na avaliação;
- **Slide 10** – Recomendações Finais da equipe de condutores.

Lista de Ações Padrões

Identificador	Ações Padrões
1	Definir guia para elaboração de casos de teste
2	Definir template para casos de teste
3	Elaborar Casos de Teste de Sistema
4	Elaborar Casos de Teste de Integração
5	Definir massa de dados
6	Definir repositório para massa de dados
7	Definir repositório para casos de teste
8	Definir critérios para priorização dos casos de teste
9	Definir atividades para automação de teste
10	Definir ferramenta para gerenciamento de Teste
11	Definir repositório de defeitos
12	Definir padrão para registro de defeitos
13	Definir guia para classificação de defeitos
14	Definir métricas básicas do produto
15	Elaborar Relatório de Teste
16	Definir atividades de inspeção dos artefatos de teste
17	Elaborar Plano de Teste
18	Elaborar scripts de automação
19	Definir relatório de acompanhamento
20	Definir controle de versão aos artefatos de teste
21	Elaborar cenários para requisitos não funcionais
22	Elaborar scripts para testes não funcionais
23	Elaborar matriz de rastreabilidade
24	Executar testes de integração
25	Executar testes de sistema
26	Executar testes de aceitação
27	Executar testes de regressão
28	Definir ferramenta para automação de teste
29	Definir ferramenta para realização dos testes não funcionais
30	Definir Plano de Métricas
31	Prover treinamentos para equipe de testes
32	Definir checklist para revisão dos requisitos
33	Realizar revisão dos requisitos
34	Definir controle de versão dos requisitos
35	Definir método de esforço e prazo
36	Definir uma equipe independente de testes
37	Definir política de testes
38	Elaborar cronograma para as atividades de teste

Priorização SWOT

Pesos das Ações para Priorização			Capacidade Estimada			
			Pontos Fortes		Pontos Fracos	
			Alta	Média	Baixa	Inexistente
Importância (em relação as metas de negócio e melhoria)	Mais Importante	Crítico	6	8	10	12
		Maior	5	7	9	11
	Menos Importante	Menor	1	2	3	4
		Irrelevante	0	0	0	0

Priorização Geral

PRIORIZAÇÃO DAS AÇÕES					
	Peso SWOT	Benefícios	Custo	Interdependência	Ordem de Prioridade
Ação 1					
Ação 2					
Ação 3					
.....					
Ação n					

Peso SWOT = Utilizar o peso da priorização resultante da técnica SWOT

Benefícios = {4- Alto, 3- Médio, 2- Irrelevante, 1- Nenhum}

Custo = {4- Nenhum, 3- Irrelevante, 2- Aceitável, 1- Nenhum}

Interdependência = {2- Independente, 1- Dependente}

Ordem de Prioridade = Somar os pesos atribuídos

Plano de Ação

Legendas:					
Status das Ações		Processos			
Ações Finalizadas		DEB_TEST	Objetivos de Debugging e Testes	ANA	Análise de Teste
Próximas Ações		PLAN	Planejamento de Teste	TEC_MET	Técnicas e Métodos de Teste
Ações Em Andamento		PROJ	Projetar Teste	CICLO	Ciclo de vida do software
		TREIN	Treinamento Teste	ACOM	Acompanhamento de Teste
Ações Planejadas		EXEC	Execução de Teste	ORG	Organização de Teste
Processo	Ação	Responsável	Observações	Data Inicial	Data Final
ITERAÇÃO 1					
	Ação 1				
	Ação 2				
	Ação n				
*PIL	Projeto Piloto				

Lições Aprendidas

INFORMAÇÕES GERAIS	
Data da Reunião	
Participantes da Reunião	<Gerente do Projeto>, <Líder Técnico>, etc...
Iteração Atual	Descreva aqui a iteração atual do programa.
	Pontos fortes relevantes
	Descrever apenas os pontos fortes relevantes. Caso não haja nenhuma, mencionar que pontos fortes significativos não foram identificados e que a equipe precisa melhorar!!
	Principais Desvios
	Listar aqui os principais desvios. Apenas os relevantes a serem apresentados para a Alta Direção e equipe do programa.
	Oportunidades de Melhoria / Observações / Lições Aprendidas relevantes
	Listar aqui as oportunidades de melhoria identificadas, observações gerais e Lições Aprendidas, se houver.

Itens de Apresentação de Resultados

- **Slide 1** – Conteúdo da Apresentação;
- **Slide 2** – Escopo da Iteração;
- **Slide 3** – Pontos Fortes da Iteração;
- **Slide 4** – Pontos de Melhoria da Iteração;
- **Slide 5** – Lições Aprendidas;
- **Slide 6** – Ações que serão integradas ao processo;

Lista de Métricas Básicas

OBJETIVOS	QUESTÕES	MÉTRICAS
Avaliar o custo de teste	Quantas horas foram gastas com as atividades de testes? Foi mais ou menos do que planejado?	Número total de horas planejadas para testes
		Número de horas para cada fase de teste
		Número de horas gastas na fase de planejamento de testes
		Número de horas gastas na fase de projeto de testes
		Número de horas gastas na fase de execução de testes
Avaliar a qualidade do produto	Quantos bugs foram detectados pela equipe de testes? Quais suas severidades?	Número de bugs de alta severidade encontrados pela equipe de testes
		Número de bugs de média severidade encontrados pela equipe de testes
		Número de bugs de baixa severidade encontrados pela equipe de testes
	Quantos bugs foram detectados pelo cliente? Quais suas severidades?	Número de bugs de alta severidade encontrados pelo cliente
		Número de bugs de média severidade encontrados pelo cliente
		Número de bugs de baixa severidade encontrados pelo cliente
Avaliar a Produtividade/Eficiência do time de testes	Qual a eficiência do time de Testes? Quantos bugs encontrados não são válidos?	Números de bugs encontrados a cada versão de teste do produto
		Número de bugs encontrados (em cada fase de teste)
		Número de caso de testes executados por hora
		Número de Bugs que não estão relacionados à nenhum caso de teste
		Número de Bugs duplicados
		Número de bugs "não reproduzíveis"
Avaliar o status das atividades de teste	Todos os casos de testes planejados foram elaborados? Quantos casos de testes foram executados? Quantos passaram? Quantos requisitos foram testados até o momento?	Número de casos de testes planejados
		Número de casos de testes elaborados até o momento
		Número de Casos de testes executados
		Número de casos de testes que passaram

	Qual a porcentagem de requisitos/features foram testados até o momento?	Número de casos de testes que falharam
		Número de casos de testes bloqueados
		Número total de requisitos/features a serem testados

Plano de Coleta de Métricas

Métrica		
Definição	Descrição:	
	Público alvo:	
Coleta	Responsável pela coleta / armazenamento:	
	Procedimento de Coleta:	
	Periodicidade:	
Armazenamento	Procedimento de Armazenamento:	
Análise	Procedimento de Análise:	

Anexo III – Questionário

Entrevistado 1 – Engenheiro de Sistemas

1- Qual é a principal diferença percebida depois da criação de um time independente de testes no projeto?

A qualidade das entregas (releases) parciais aumentou consideravelmente, tanto em relação ao produto (software) em si, como a percepção do time em relação ao problema do cliente.

2- Como foi a integração entre o time de testes e desenvolvimento?

Inicialmente, houve uma separação entre o time de desenvolvimento e o time de testes, devido, principalmente, aos conflitos de idéias e entendimento dos requisitos. Porém, ao longo das releases, passou-se a ter uma integração mais construtivas entre as duas partes.

3- Qual foi o principal benefício da integração do time de testes com o ciclo de vida do produto?

Percepção dos requisitos do cliente, aumentando a qualidade das entregas parciais.

4- A participação dos engenheiros de testes na revisão dos requisitos ajudou ou prejudicou a qualidade final do produto?

Mesmo comentário da questão anterior.

5- Na sua opinião, o que poderia ser melhorado nas atividades de testes como um todo, para tornar mais efetiva a atuação dos engenheiros de testes na qualidade final do produto?

O uso de outras abordagens de testes existentes no mercado: testes de código (jUnits), testes de estresse, testes automáticos, dentre outros. Outro ponto a ser considerado é que nem todos os engenheiros de testes tomaram o cuidado de estudar bem o requisito do cliente, levando a muitas vezes a abertura de correção de bugs sem sentido ou que agregavam valor ao negócio do cliente.

- 6- Como a inserção de práticas ágeis no processo de testes pode ter contribuído ou prejudicado o desempenho das atividades de testes?**

Contribuiu consideravelmente.

- 7- Quais são as principais dificuldades e restrições para o desempenho das atividades de testes?**

Cronograma e tempo de testes após a entrega da funcionalidade por parte dos engenheiros de desenvolvimento.

- 8- Na sua opinião, as mudanças que existiram durante o projeto na área de testes foram positivas? Qual a mudança que trouxe maior benefício para o projeto?**

Altamente positivas, e as principais mudanças (como já falado) foram a melhor percepção dos requisitos do cliente e a melhoria (considerável) das entregas parciais do produto.

Entrevistado 2 – Líder Técnico do Time de Desenvolvimento

1- Qual é a principal diferença percebida depois da criação de um time independente de testes no projeto?

Foi observada uma melhoria considerável da qualidade do produto, assim como do código produzido. Além disso, o entendimento dos requisitos se tornou mais eficaz, visto que estavam sempre sendo revistos e validados com os testes.

2- Como foi a integração entre o time de testes e desenvolvimento?

No começo foi um pouco difícil para algumas pessoas. Os desenvolvedores não estavam acostumados a ter um Engenheiro de Testes criticando a funcionalidade implementada por ele. Da mesma forma, os engenheiros de testes estavam começando a testar uma aplicação ainda não testada, ou seja, com muitos erros básicos. Outra dificuldade foi o entendimento dos requisitos, que muitas vezes divergiam entre as duas equipes.

3- Qual foi o principal benefício da integração do time de testes com o ciclo de vida do produto?

Maior garantia de qualidade do produto. As entregas feitas passaram a ser estáveis. Outra vantagem se deu na elicitación e evolução dos requisitos da aplicação.

4- A participação dos engenheiros de testes na revisão dos requisitos ajudou ou prejudicou a qualidade final do produto?

Ajudou bastante, visto o intenso contato que o engenheiro de testes tem com a aplicação. O tester é o papel mais próximo do verdadeiro do usuário da aplicação.

5- Na sua opinião, o que poderia ser melhorado nas atividades de testes como um todo, para tornar mais efetiva a atuação dos engenheiros de testes na qualidade final do produto?

Os resultados das atividades de testes, os casos de testes, métodos utilizados,... estas informações deveriam ser melhor compartilhadas com os desenvolvedores, para que estes pudessem melhorar também seus testes unitários. Outro ponto que pode ser melhorado é a automação de testes, o que pode aumentar a capacidade de execução de testes da equipe.

6- Como a inserção de práticas ágeis no processo de testes pode ter contribuído ou prejudicado o desempenho das atividades de testes?

Creio que os benefícios e prejuízos são os mesmos observados nas atividades de desenvolvimento.

7- Quais são as principais dificuldades e restrições para o desempenho das atividades de testes?

Creio que a maior dificuldade se encontra na garantia dos testes feitos entre diferentes versões da aplicação. Muitas vezes não é possível garantir que todos os testes executados com sucesso numa versão, vão executar da mesma forma numa versão seguinte. Outra dificuldade é a documentação dos resultados de testes e a integração desta informação com as CRs (Change Requests) originadas de cada teste. Este registro sempre é muito trabalhoso.

8- Na sua opinião, as mudanças que existiram durante o projeto na área de testes foram positivas? Qual a mudança que trouxe maior benefício para o projeto?

Foram muito positivas. As mudanças que trouxeram maiores benefícios foram duas: Primeiramente o envolvimento da equipe de testes na execução de testes de forma continua, durante todo o período da Sprint. Outra vantagem foi o envolvimento da equipe de testes na revisão e evolução dos requisitos.

Entrevistado 3 – Engenheiro de Testes

1- Qual é a principal diferença percebida depois da criação de um time independente de testes no projeto?

A qualidade do software melhorou muito em vários aspectos.

2- Como foi a integração entre o time de testes e desenvolvimento?

Em principio percebi certa resistência da equipe de desenvolvimento em acatar as solicitações de alteração tornando a integração difícil.

3- Qual foi o principal benefício da integração do time de testes com o ciclo de vida do produto?

O principal benefício foi, resultados mais próximos da expectativa do cliente, com melhorias crescentes no sistema.

4- A participação dos engenheiros de testes na revisão dos requisitos ajudou ou prejudicou a qualidade final do produto?

Ajudou na qualidade do documento e com isso a melhora do produto.

5- Na sua opinião, o que poderia ser melhorado nas atividades de testes como um todo, para tornar mais efetiva a atuação dos engenheiros de testes na qualidade final do produto?

Testes executados no código e caso de testes baseados em casos de uso criando para tentar criar uma rastreabilidade das alterações aproximando o teste de regressão do necessário para melhorar o tempo de execução dos testes.

6- Como a inserção de práticas ágeis no processo de testes pode ter contribuído ou prejudicado o desempenho das atividades de testes?

Houve melhoria, pois com as atividades bem definidas a execução e os resultados são alcançados de forma eficaz.

7- Quais são as principais dificuldades e restrições para o desempenho das atividades de testes?

Como o sistema é grande o tempo para a execução dos testes deveria ter maior prazo para execução.

8- Na sua opinião, as mudanças que existiram durante o projeto na área de testes foram positivas? Qual a mudança que trouxe maior benefício para o projeto?

Sim, pois os testes foram progredindo e amadurecendo juntamente com o sistema tornando uma melhoria considerável na qualidade. Testes de performance, melhorias nos documentos casos de teste.

Entrevistado 4 – Engenheiro de Testes

1- Qual é a principal diferença percebida depois da criação de um time independente de testes no projeto?

A entrega dos produtos com qualidade, o que não acontecia anteriormente.

2- Como foi a integração entre o time de testes e desenvolvimento?

Um pouco desgastante até que os dois times entendessem os papéis e objetivos de cada um.

3- Qual foi o principal benefício da integração do time de testes com o ciclo de vida do produto?

A qualidade do produto.

4- A participação dos engenheiros de testes na revisão dos requisitos ajudou ou prejudicou a qualidade final do produto?

Ajudou, uma vez que os engenheiros de teste identificaram erros e falta de informações na escrita dos requisitos, o que ocasionou alterações que clarificaram o entendimento dos mesmos.

5- Na sua opinião, o que poderia ser melhorado nas atividades de testes como um todo, para tornar mais efetiva a atuação dos engenheiros de testes na qualidade final do produto?

Uma melhor análise dos resultados, a fim de identificar os defeitos escapados e assim, efetuar uma melhor escolha dos testes para execução.

6- Como a inserção de práticas ágeis no processo de testes pode ter contribuído ou prejudicado o desempenho das atividades de testes?

Contribuiu, uma vez que o acompanhamento das atividades foi realizado de forma efetiva e eficiente, fazendo com que os prazos fossem repensados com tempo hábil.

7- Quais são as principais dificuldades e restrições para o desempenho das atividades de testes?

A falta de tempo nas entregas dos produtos, para análise de resultados e para treinamento de novos engenheiros de teste.

8- Na sua opinião, as mudanças que existiram durante o projeto na área de testes foram positivas? Qual a mudança que trouxe maior benefício para o projeto?

Certamente foram positivas. A mudança que trouxe maior benefício para o projeto foi a qualidade do produto final junto ao cliente.

Entrevistado 5 – Representante 1 do Cliente

1- Qual é a principal diferença percebida depois da criação de um time independente de testes no projeto?

R.: A qualidade da entrega

2- A participação dos engenheiros de testes na revisão dos requisitos ajudou ou prejudicou a qualidade final do produto?

R.: Ajudou muito pois melhorou consideravelmente o entendimento das regras de negócio entre a equipe do projeto.

3- Em sua opinião, o que poderia ser melhorado nas atividades de testes como um todo, para tornar mais efetiva a atuação dos engenheiros de testes na qualidade final do produto?

R: Estimular o envolvimento dos engenheiros de testes com os usuários o mais cedo possível no projeto.

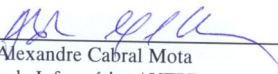
4- Em sua opinião, as mudanças que existiram durante o projeto na área de testes foram positivas? Qual a mudança que trouxe maior benefício para o projeto?

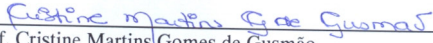
R. Extremamente positivas. Quando houve o envolvimento dos engenheiros de testes na revisão dos requisitos.

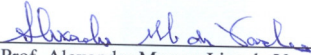
Entrevistado 6 – Representante 2 do Cliente

- 9- **Qual é a principal diferença percebida depois da criação de um time independente de testes no projeto?** A qualidade dos produtos (releases) entregue após a criação do time de testes melhorou muito.
- 10- **A participação dos engenheiros de testes na revisão dos requisitos ajudou ou prejudicou a qualidade final do produto?** Melhorou
- 11- **Em sua opinião, o que poderia ser melhorado nas atividades de testes como um todo, para tornar mais efetiva à atuação dos engenheiros de testes na qualidade final do produto?** O tempo de execução dos testes.
- 12- **Em sua opinião, as mudanças que existiram durante o projeto na área de testes foram positivas? Sim Qual a mudança que trouxe maior benefício para o projeto?** A participação mais ativa da equipe de testes na consolidação dos requisitos passou uma segurança maior nos resultados dos testes.

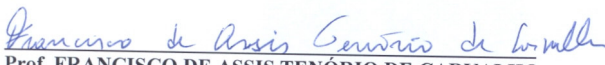
Dissertação de Mestrado apresentada por **Juliana Ochner** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**Uma Abordagem de Implantação de Testes Baseada em Metodologias Ágeis**", orientada pelo **Prof. Alexandre Marcos Lins de Vasconcelos** e aprovada pela Banca Examinadora formada pelos professores:


Prof. Alexandre Cabral Mota
Centro de Informática / UFPE


Prof. Cristine Martins Gomes de Gusmão
Departamento de Sistemas Computacionais / UPE


Prof. Alexandre Marcos Lins de Vasconcelos
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 28 de fevereiro de 2008.


Prof. FRANCISCO DE ASSIS TENÓRIO DE CARVALHO
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Ochner, Juliana

Uma abordagem para implantação de testes baseada em metodologias ágeis / Juliana Ochner. – Recife : O Autor, 2008.

xi, 173 folhas : il., fig., tab.

Dissertação (mestrado) – Universidade Federal de Pernambuco. Cln. Ciência da Computação, 2007.

Inclui bibliografia e anexos.

1. Engenharia de software. I. Título.

005.1 CDD (22.ed.) MEI2008-053