

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
UNIDADE ACADÊMICA DE EDUCAÇÃO CONTINUADA
ESPECIALIZAÇÃO EM QUALIDADE DE SOFTWARE

Pamela Silva da Silva

AVALIAÇÃO DO PROCESSO DE TESTE DE SOFTWARE DA ORGANIZAÇÃO ABC
UTILIZANDO O *TEST PROCESS IMPROVEMENT*

São Leopoldo

2015

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
UNIDADE ACADÊMICA DE EDUCAÇÃO CONTINUADA
ESPECIALIZAÇÃO EM QUALIDADE DE SOFTWARE

Pamela Silva da Silva

AVALIAÇÃO DO PROCESSO DE TESTE DE SOFTWARE DA ORGANIZAÇÃO ABC
UTILIZANDO O *TEST PROCESS IMPROVEMENT*

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do título de Especialista em Qualidade de Software, pelo curso de Pós-Graduação Lato Sensu em Qualidade de Software da Universidade do Vale do Rio dos Sinos – UNISINOS.

Orientadora: Prof. Esp. Priscila Coelho Blauth

São Leopoldo

2015

Avaliação do Processo de Teste de Software da Organização ABC utilizando o *Test Process Improvement*

Pamela Silva da Silva¹

¹Universidade do Vale do Rio dos Sinos (Unisinos)
São Leopoldo – RS – Brasil

pamelacharq@gmail.com

Abstract. *The search for quality products has made organizations invest in testing and software development. To ensure the software quality, it must have a well-defined testing process with trained professionals. This article presents an evaluation of software testing process in the ABC Organization through a case study using the Test Process Improvement model - TPI and suggestions for improvements. The results showed that the Organization does not have high test maturity level and needs to improve its testing process to have a quality end product.*

Resumo. *A busca pela qualidade dos produtos tem feito as organizações investirem em desenvolvimento e teste de software. Para garantir a qualidade do software, é necessário ter um processo de teste bem definido com profissionais capacitados. Este artigo apresenta uma avaliação do processo de teste de software da Organização ABC através de um estudo de caso, utilizando o modelo Test Process Improvement – TPI e sugestões de melhorias. Os resultados mostraram que a Organização não tem o nível de maturidade de teste elevado e precisa melhorar seu processo de teste para ter um produto final de qualidade.*

1. Introdução

Atualmente as organizações passaram a investir em desenvolvimento de software devido ao aumento crescente de novas tecnologias. Para Bastos (2007), o objetivo principal do processo de teste é simplesmente encontrar o maior número possível de defeitos no software, ou seja, através dos processos de teste de software, buscar melhorias e qualidade nos produtos para aumentar a eficiência e diminuir custos.

A exigência da qualidade do software gerou a necessidade por melhorias contínuas no processo de desenvolvimento. A garantia da qualidade estabelece a infraestrutura que suporta métodos sólidos de engenharia de software, gerenciamento racional de projeto e ações de controle de qualidade, todos fundamentais para a construção de software de alta qualidade (PRESSMAN, 2011). A garantia da qualidade e o processo de teste de software auxiliam as Organizações a oferecerem produtos qualificados, com requisitos em conformidade e que atendam as necessidades do usuário. A qualidade do software está diretamente relacionada com a qualidade do processo de desenvolvimento e com o processo de teste.

Quando se trata de testes, também se refere à qualidade do software, pois o nível de qualidade dos testes de um software é um fator importante, entre outros, para definir a qualidade do produto final, que depende do processo de desenvolvimento desse software (RIOS, 2013).

A motivação para o desenvolvimento deste artigo é avaliar o quanto o processo de teste da Organização em estudo está adequado e definido. Um dos fatores que levaram a aplicar a avaliação do processo de teste é o nível de qualidade que o sistema está sendo entregue para o cliente que muitas vezes não é satisfatório. A falta de padronizações no processo de teste e treinamentos para os testadores prejudica na qualidade final do produto. A justificativa para avaliar o processo de teste da Organização é a busca pela melhoria contínua dos processos e da qualidade do sistema para atender as exigências dos clientes, utilizando o modelo TPI, o mais adequado e completo para a Organização. Quando não há um processo de teste bem definido, os testes tornam-se incompletos durante o desenvolvimento e poderão ocorrer problemas após a implantação do software.

Para Pressman (2011), os custos relativos para descobrir e reparar um erro ou defeito aumentam drasticamente à medida que avançamos dos custos de prevenção para custos de detecção de falhas internas e para custos de falhas externas, ou seja, o custo para se corrigir um defeito no software em produção é muito alto e pode causar prejuízos para a empresa e para o cliente.

Para avaliar o processo de teste da Organização, será utilizado o modelo *Test Process Improvement* – TPI. Este modelo é o que mais se adapta às Organizações e é direcionado para uma avaliação dos níveis de maturidade dos processos de testes. Segundo Koomen (1999), o modelo foi desenvolvido baseado no conhecimento prático e em experiências em processos e melhores práticas de testes de software. Por estar mais próximo da realidade das Organizações, o modelo é fácil de ser aplicado e possui as áreas mais importantes do processo de teste que precisam ser avaliadas.

O modelo TPI apresenta uma visão da maturidade dos processos de testes da Organização e trabalha com 20 principais áreas chaves, dentre elas, Estratégia de teste, Modelo do ciclo de vida e Planejamento e estimativa que conduzem à classificação dos processos de testes. No processo de avaliação são utilizados pontos de verificações por áreas chaves, o relacionamento entre os níveis na matriz de maturidade de testes e as sugestões de melhorias (KOOMEN, 1999). Melhorar o processo de testes implica em melhorar a qualidade e confiança dos produtos.

O artigo tem com objetivo geral avaliar o processo de teste de uma Organização utilizando o modelo *Test Process Improvement* – TPI, analisar os resultados e propor melhorias para alcançar o nível desejado e ter um processo de teste mais adequado e definido.

Com base na avaliação do processo de teste serão identificados os pontos fortes e fracos do processo de teste da Organização. Através da análise do resultado da avaliação, serão exibidos os níveis de maturidade das áreas chaves do processo de teste que a empresa se encontra atualmente. Sugestões de melhorias serão propostas para que o processo de teste alcance o nível de maturidade esperado e melhore os processos. Para atingir o objetivo geral, estes foram os objetivos específicos definidos.

Na avaliação do processo de teste da Organização, não será utilizado como base outros modelos de maturidade de testes existentes, somente o modelo *Test Process Improvement* – TPI. A avaliação não será aplicada com todos os Analistas de testes da Organização, mas sim com Analistas de testes que pertencem a uma equipe de desenvolvimento na qual a autora do trabalho também atua como Analista de testes, e com isso ter mais facilidades de comunicação.

Com a avaliação do processo de teste, pretende-se propor sugestões para melhorar o processo de teste atual da Organização, padronizar os processos de teste através de técnicas, metodologias e ter profissionais treinados e capacitados. Estes fatores contribuem para ter um produto de qualidade ao longo do ciclo de vida de desenvolvimento do software. O *Test Process Improvement* – TPI foi utilizado por ser muito flexível e completo, e para contribuição em pesquisas para a área de teste de software sobre o modelo, pois não há uma grande quantidade de materiais disponíveis para estudo.

Este trabalho está organizado em cinco capítulos. No capítulo 1, é apresentada a Introdução, com a motivação e justificativa para o desenvolvimento deste estudo, os objetivos, limitações do tema e como ele está organizado. O capítulo 2 apresentará a fundamentação teórica para o desenvolvimento deste trabalho, apresentando conceitos sobre processo de teste, ciclo de vida do processo de teste, modelo "V", alguns modelos de maturidade de teste, detalhando o *Test Process Improvement* – TPI modelo utilizado neste trabalho, e trabalhos relacionados. No capítulo 3, será apresentada a metodologia, ou seja, qual método de pesquisa utilizado, a população alvo do estudo, quais técnicas de coleta e análise serão utilizadas, as limitações do estudo, e por fim, as etapas desenvolvidas. No capítulo 4, será apresentado o estudo de caso realizado, ou seja, a execução da avaliação do processo de teste na Organização utilizando o modelo *Test Process Improvement* – TPI, análise dos resultados da avaliação, proposta e validação das melhorias. E, por fim, o capítulo 5 apresenta as conclusões obtidas através do estudo realizado.

2. Fundamentação teórica

2.1. Processo de teste

O teste, da maneira como é executado pela maioria das empresas, como uma etapa dentro do processo de desenvolvimento e, em geral, executado pelos próprios desenvolvedores e pelos usuários do sistema, serve apenas para garantir que as especificações ou os requisitos do sistema foram de fato implementados (BASTOS, 2007). Em um modelo de garantia da qualidade isto é insuficiente, pois somente com testes unitários não é possível garantir que o software esteja livre de erros. Para ter um software final com qualidade é necessário ter um processo de teste definido e com profissionais capacitados na área.

Defeitos existentes no software constituem-se em riscos tanto para o negócio quanto para a imagem da empresa. Com um processo de teste, uma metodologia adequada e equipe de testes especializada é possível minimizar os riscos causados por defeitos. No início do projeto de desenvolvimento do software, através de revisões e

inspeções, os testes podem ser feitos no documento de requisitos, pois a correção no início do projeto custa menos do que encontrar defeitos em produção.

A Figura 1 baseada em Bastos (2007), mostra a Regra 10 de Myers, que estabelece que o custo da correção de defeitos tende a aumentar quanto mais tarde o defeito for detectado. Defeitos encontrados durante a produção tendem a custar muito mais que defeitos encontrados em modelos de dados e em outros documentos do projeto de software.

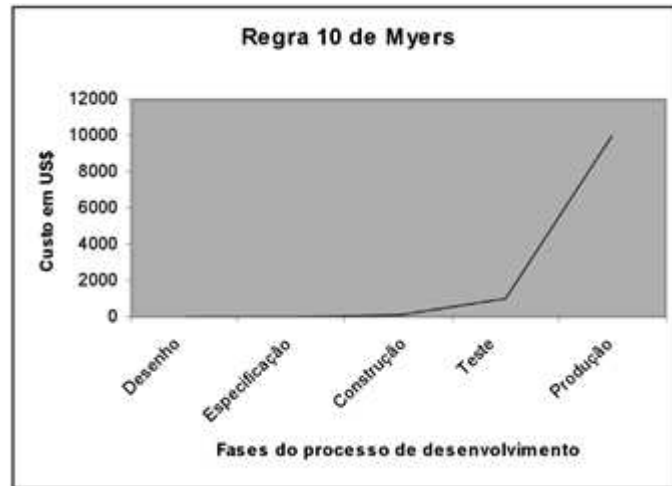


Figura 1. Regra 10 de Myers

Segundo Rios (2013), o ciclo de vida do processo de testes mostrado na Figura 2, é composto por diversas etapas ou fases, sendo quatro delas sequenciais ou em cascata e duas paralelas.

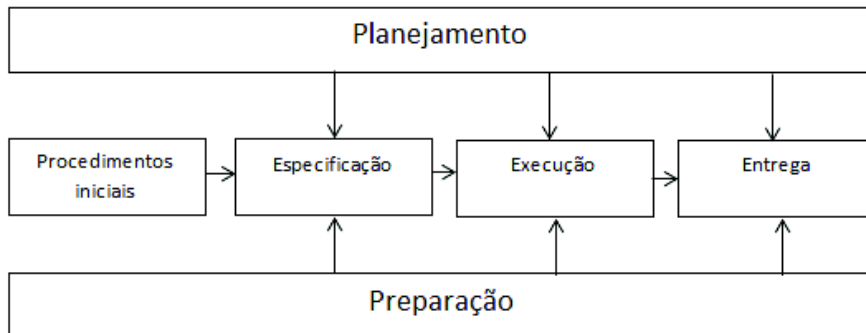


Figura 2. Ciclo de vida do processo de testes

Este modelo é chamado de Modelo 3P x 3E e cada uma de suas etapas correspondem a atividades, produtos e documentos.

Procedimentos iniciais: Nesta etapa é elaborado um plano com todas as atividades principais que serão executadas, recursos de pessoal e de ambiente e um estudo dos requisitos do negócio do sistema que será desenvolvido.

Planejamento: A Estratégia de Teste e o Plano de Teste são elaborados para minimizar os riscos do negócio. Esta etapa deve ser executada em conjunto com as atividades de captação de requisitos e planejamento do projeto com testes de verificação sobre os requisitos. A atividade de planejamento deve permanecer ativa até a conclusão do projeto.

Preparação: O objetivo desta etapa é preparar o ambiente de teste (equipamentos, pessoal ferramentas, hardware e software) para que os testes sejam executados.

Especificação: Nesta etapa são elaborados os casos de testes e roteiros de teste.

Execução: Executar os casos de teste e registrar os resultados.

Entrega: O projeto de teste é finalizado, são relatadas as ocorrências do projeto que foram consideradas relevantes à melhoria de processo.

O processo de testes depende dos produtos gerados nas fases do ciclo de desenvolvimento e que, apesar de serem independentes, tem início simultaneamente. O ciclo de vida de testes pressupõe que sejam realizados testes durante todo o processo de desenvolvimento (BASTOS, 2007). A equipe de desenvolvimento inicia o processo de desenvolvimento do sistema e a equipe de testes inicia o processo de planejamento do processo de teste, utilizando as mesmas informações que estão no documento de requisitos.

O modelo “V” exhibe as fases individuais de teste e desenvolvimento em forma de “V”: Verificação e Validação. Verificação e validação (V&V) é o nome dado aos processos de verificação e análise que asseguram que o software cumpra suas especificações e atenda às necessidades do cliente (SOMMERVILLE, 2005), isto é, verificar se o que foi documentado está implementado corretamente no software. Segundo Pressman (2011), a verificação refere-se ao conjunto de tarefas que garantem que o software implementa corretamente uma função específica. A validação refere-se a um conjunto de tarefas que asseguram que o software foi criado e pode ser rastreado segundo os requisitos do cliente.

2.2. Modelos de maturidade em teste de software

As Organizações procuram cada vez mais melhorar os seus processos de desenvolvimento de software com foco na melhoria da qualidade dos produtos. Os modelos de referência CMMI - *Capability Maturity Model Integration* e MPS. Br - Melhoria de Processos do Software Brasileiro, são amplamente utilizados para melhorar o processo de desenvolvimento, mas não dão muita ênfase aos testes, são descritas somente práticas que estão relacionadas com as atividades de Verificação e Validação.

Os modelos de melhoria de processo de teste foram criados com o objetivo de serem complementares aos modelos de processo de desenvolvimento, dando destaque ao teste de software. Ao utilizar um modelo de maturidade de processo, é possível obter informações do que deve conter um processo bem definido e as melhores práticas existentes para ter subsídio para avaliação da maturidade do processo. Os modelos de maturidade surgiram para avaliar e melhorar o nível de qualidade dos processos de testes aplicados numa Organização desenvolvedora de software. Alguns dos modelos de testes desenvolvidos são: *Test Process Improvement – TPI*, *Test Maturity Model Integrated – TMMI* e Melhoria do Processo de Teste Brasileiro – MPT.BR.

2.2.1. Test Process Improvement – TPI

O *Test Process Improvement – TPI* é um modelo de maturidade de teste desenvolvido por Tim Koomen e Martin Pol, em 1997. O modelo é baseado no conhecimento, experiências em processos e melhores práticas de testes de software. Este possibilita avaliar os níveis de maturidade do processo de testes atual de uma Organização possibilitando a melhoria contínua das atividades e controlando gradualmente as

melhorias necessárias para alcançar os níveis mais elevados na sua implementação (KOOMEN, 1999).

O modelo TPI é utilizado para analisar a situação atual do processo de teste da Organização e mostrar os pontos fortes e fracos de diferentes pontos de vista, como por exemplo, o uso de Ferramentas de teste, Técnicas de especificação de teste e Relatórios, chamados de Áreas chave. Com a avaliação de cada área chave é possível classificar o processo de teste em níveis de maturidade que podem ser de A até D.

Os níveis e as áreas chaves estão relacionados uns aos outros em uma matriz de maturidade de teste (*Test Maturity Matrix*) e para cada nível são atribuídos pontos de verificação (*checkpoints*). Os pontos de verificação são requisitos, se o processo de teste atingiu todos os pontos de verificação de um certo nível, então o processo é classificado para aquele nível. Sugestões de melhoria são propostas com instruções para alcançar o nível desejado.

Segundo Koomen (1999), o TPI possui uma estrutura bem definida conforme a Figura 3:

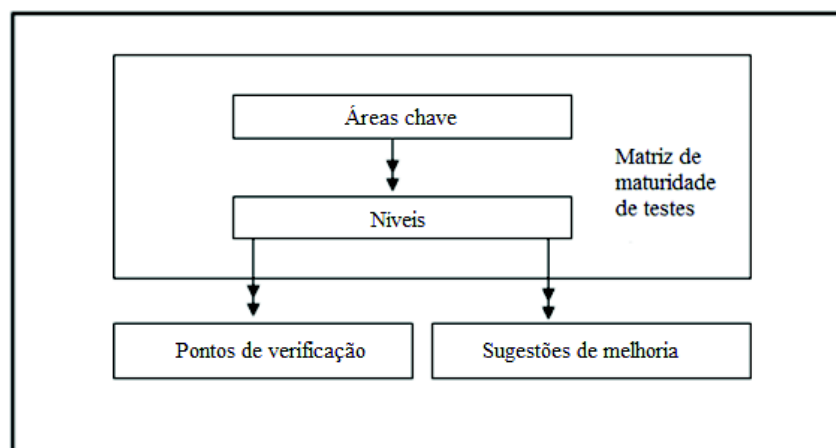


Figura 3. Estrutura TPI

O modelo TPI, baseia-se na metodologia de teste de software *Test Management Approach* – TMAP. Esta metodologia descreve um processo estruturado de testes e permite organizar os processos de testes em Organizações que desejam alcançar níveis de qualidade dos seus produtos.

O TMAP possui quatro pilares que são: Ciclo de vida, Organização, Infraestrutura e ferramentas e Técnicas. Os pilares referem-se ao ciclo de vida das atividades de teste relacionadas com o ciclo de vida de desenvolvimento, boa Organização, infraestrutura e ferramentas adequadas e técnicas utilizadas para realizar as atividades de teste. A implementação do TMAP não é um objetivo final do modelo TPI, mas os detalhes da metodologia são usados para fornecer um modelo mais completo possível (KOOMEN, 1999).

2.2.1.1. Áreas chave do TPI

O modelo TPI possui 20 áreas chaves estruturadas em quatro grupos, que cobrem todo o processo de testes que são: Ciclo de vida, Organização, Infraestrutura e ferramentas e Técnicas (KOOMEN, 1999).

As 20 áreas chaves do TPI são descritas abaixo:

- Grupo Ciclo de vida

Estratégia de teste: A Estratégia de teste é uma das áreas chaves mais importantes do TPI, pois são definidos quais requisitos e riscos serão cobertos pelo processo de testes. Tem por objetivo identificar as exigências e os riscos da qualidade do produto para a Organização. Apresenta como foco a detecção dos defeitos mais importantes o mais cedo e com menor custo possível.

Modelo de ciclo de vida: No ciclo de vida do processo de testes existem várias fases como: planejamento, preparação, especificação, execução e entrega. Cada fase consiste em um número de atividades com determinados objetivos e produtos a serem entregues, como por exemplo: proposta, entradas, o processo, saídas, as dependências, as técnicas e ferramentas aplicadas e documentação. A utilização de um modelo de ciclo de vida possibilita prever e controlar as melhorias, atividades de planejamento e controle do processo de testes.

Momento do envolvimento: O envolvimento do teste deve ser o mais cedo possível, antes do software ser desenvolvido. O envolvimento tardio dos testes é um risco para o projeto, pois os defeitos são mais caros e mais difíceis de serem corrigidos. Quanto mais cedo for o envolvimento dos testes, melhor é para detecção dos defeitos e correção no início do processo de desenvolvimento do software.

- Grupo Técnicas

Planejamento e estimativa: O planejamento e estimativa indicam quais atividades devem ser realizadas, quais recursos (pessoas) são necessários para a execução das atividades de testes no período de tempo previsto.

Técnicas de especificação de testes: A técnica de especificação de testes é uma forma padronizada na obtenção dos casos de testes a partir de fontes de informação (entrevistas, documentações).

Técnicas estáticas de teste: Inspeccionar produtos sem execução de programas por meio de *Checklists* e com certo grau de qualidade é chamado de testes estáticos. A avaliação pode ser realizada nas documentações do sistema.

Métricas: Métricas são observações quantificadas das características do produto ou processo. Para o processo de teste, as métricas do progresso do processo de teste e a qualidade do sistema testado são muito importantes. São utilizadas para controlar o processo de teste e para comparar sistemas ou processos. As métricas são importantes para avaliar a consequência de certas ações de melhorias comparando os dados antes e depois da implementação da melhoria.

- Grupo Infraestrutura e ferramentas

Ferramentas de teste: A automação contribui para a execução das atividades de teste como planejamento, controle, especificação, execução e análise em menos tempo e obter maior produtividade e assim realizar testes mais profundos. Dessa forma, é possível executar uma grande quantidade de testes com um mínimo de esforço.

Ambiente de Testes: A execução dos testes ocorre no chamado ambiente de teste. Este ambiente é composto por hardware, softwares de apoio, formas de comunicação,

facilidade de montar ou utilizar banco de dados e possui grande influência na qualidade, tempo e custo do processo de teste.

Local de Trabalho: O local de trabalho da equipe de testes tem uma forte influência no rendimento da equipe. É importante que os profissionais tenham equipamentos adequados para desempenhar as atividades como, por exemplo: mesas, cadeiras, computadores, softwares, telefone, etc.

- Grupo Organização

Motivação e comprometimento: O comprometimento e motivação do pessoal envolvido no teste é um pré-requisito importante para a fácil execução das atividades de teste. O processo de teste deve possuir tempo e recursos suficientes para executar um bom teste, no qual a cooperação e boa comunicação com o resto do projeto resultam num processo com ótima eficiência. Um ambiente de testes organizado e estruturado é uma grande motivação para a equipe de teste.

Atividades e treinamentos de testes: Em um processo de teste, a equipe treinada e capacitada é um fator importante para o desempenho das atividades. Diferentes formações, conhecimentos e habilidades dos profissionais são fundamentais. Além da experiência, é necessário conhecer o negócio e o produto da Organização.

Escopo da metodologia: Para cada processo de teste na Organização, uma certa metodologia ou método de trabalho é utilizado, composto por atividades, procedimentos, regulamentos, técnicas e etc. O objetivo é utilizar uma metodologia suficientemente genérica, aplicável em qualquer situação, mas que contenha detalhes suficientes para que não seja necessário repensar os mesmos itens novamente a cada vez que for utilizada.

Comunicação: Num processo de teste a comunicação com todas as pessoas envolvidas, testadores, desenvolvedores, usuários e clientes é muito importante. Com a comunicação todos ficarão informados sobre o progresso e qualidade do produto.

Relatórios: Os relatórios são fundamentais na identificação das falhas e melhorias na qualidade do produto. O objetivo principal é fornecer critérios de avaliação fundamentada do produto e do processo de desenvolvimento.

Gerenciamento dos defeitos: Um bom gerenciamento deve ser capaz de mostrar o ciclo de vida dos defeitos, suporte à análise e resolução dos mesmos.

Gerenciamento dos produtos de teste: Os produtos de teste como Planos de testes, casos de testes, especificações, bases de testes e relatórios devem ser controlados e administrados corretamente para que possam ser reutilizados em projetos semelhantes. Através do gerenciamento, é possível identificar anomalias causadas por uma versão errada do sistema.

Gerenciamento do processo de teste: Para gerenciar cada processo e atividade de teste, os quatro passos do ciclo Deming são essenciais: Planejar (Plan), Executar (Do), Verificar (Check) e Agir (Act). O gerenciamento do processo é de vital importância para a realização de um teste ótimo num processo de teste frequentemente turbulento.

O PDCA é também conhecido como ciclo de *Shewhart* ou ciclo de *Deming*, é um método de desenvolvimento que tem foco na melhoria contínua (AMORIM, 2015). O ciclo de Deming é constituído por quatro etapas, apresentadas a seguir:

Plan: Elaborar um plano de ação estabelecendo os objetivos a serem alcançados, o caminho a ser traçado até que o objetivo seja atingido e definir o método que deve ser utilizado para consegui-los.

Do: É a parte da execução propriamente dita e o seu acompanhamento com base em relatórios para uma posterior análise.

Check: Monitorar e avaliar periodicamente os resultados, avaliar processos e confrontá-los com o resultado planejado.

Act: Agir de acordo com o avaliado e os seus relatórios, e com base nessas informações, se necessário, determinar novos planos de ação, de forma a melhorar a qualidade, eficiência e eficácia, corrigindo eventuais falhas.

- Grupo Todos

Avaliação: A avaliação dos produtos intermediários do desenvolvimento refere-se ao processo de inspeção dos requisitos funcionais ou negócios da Organização com objetivo de encontrar defeitos em estágios iniciais reduzindo os custos de retrabalhos.

Testes de baixo nível: Os testes de baixo nível são os testes executados diretamente nos códigos fontes pelo próprio desenvolvedor (testes unitários e integração). Tem por objetivo encontrar defeitos ainda no processo de desenvolvimento do produto, evitando que os defeitos sejam localizados nos testes de alto nível.

2.2.1.2. Níveis de maturidade TPI

Os níveis de maturidade do TPI para cada área chave são divididos em quatro: A, B, C e D. O número de níveis não é o mesmo para cada área chave. Para atingir um nível, os pontos de verificação que são os requisitos, devem ser alcançados. A classificação dos níveis ocorre através da revisão dos requisitos dos níveis anteriores, para avançar um nível é necessário que todos os requisitos do nível anterior sejam respeitados.

O processo de teste classificado como nível B de uma certa área chave, alcançou os pontos de verificação do nível A e do nível B. Quando o processo de teste não alcança os pontos de verificação do nível A, o processo está no nível Inicial e na escala zero na matriz de maturidade.

2.2.1.3. Matriz de maturidade

O objetivo principal da matriz de maturidade de testes é avaliar os relacionamentos entre as diferentes áreas chaves e os níveis de maturidade e determinar ações de melhorias alcançando níveis mais elevados dentro do processo de testes. A matriz de maturidade possui uma escala de 13 itens e 3 níveis de melhorias, conforme o Quadro 1 (KOOMEN, 1999). Os níveis de melhorias são: Controlado, Eficiente e Otimizado. A evolução do processo de teste é representada através da escala e dos níveis, quanto maior o nível dentro da escala de maturidade, melhor é para o processo de testes.

Há uma dependência entre alguns níveis e áreas chaves, mas não interferem na decisão de uma área chave alcançar um certo nível.

Quadro 1. Matriz de Maturidade

Escalas	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Áreas chave	Inicial	Controlado				Eficiente				Otimizado				
1 Estratégia de teste		A					B				C		D	
2 Modelo de ciclo de vida		A			B									
3 Momento do envolvimento			A				B				C		D	
4 Planejamento e estimativa				A							B			
5 Técnicas de especificação de testes		A		B										
6 Técnicas estáticas de teste					A		B							
7 Métricas						A			B			C		D
8 Ferramentas de teste					A			B			C			
9 Ambiente de testes				A				B						C
10 Local de trabalho				A										
11 Motivação e comprometimento		A				B						C		
12 Atividades e treinamentos de testes				A			B				C			
13 Escopo da metodologia					A						B			C
14 Comunicação			A		B							C		
15 Relatórios		A			B		C					D		
16 Gerenciamento dos defeitos		A				B		C						
17 Gerenciamento dos produtos de teste			A			B				C				D
18 Gerenciamento do processo de teste		A		B								C		
19 Avaliação							A			B				
20 Testes de baixo nível					A		B		C					

As categorias das escalas de maturidade são descritas abaixo:

Controlado - As escalas da maturidade de 1 a 5 têm foco no controle do processo de teste. O propósito dos níveis é proporcionar um processo de teste controlado e ter uma visão da qualidade do sistema testado. O processo do teste tem uma estratégia previamente definida, técnicas de especificação de teste, os defeitos são registrados e relatados. Os produtos e ambiente de teste estão bem controlados e os testadores são adequadamente treinados.

Eficiente - Nas escalas da maturidade de 6 a 10, o foco está na eficiência do processo de testes. Esta eficiência é alcançada através de uma melhor integração entre os processos de teste e outras partes do desenvolvimento do sistema e ao consolidar o método de trabalho do processo de teste na Organização.

Otimizado - Nas escalas da maturidade de 11 a 13, tem foco no aumento e na otimização do processo de testes, ou seja, visa assegurar melhoria contínua do processo de testes da Organização.

2.2.1.4. Pontos de verificação

Os pontos de verificação (*checkpoints*) são instrumentos de medição para determinar o nível de uma área chave no processo de teste. Cada nível de uma área chave tem pontos de verificação que precisam ser alcançados para que seja classificado naquele nível. Com base nos *checkpoints*, um processo de teste pode ser avaliado. Cada nível seguinte de uma área-chave corresponde uma melhoria que orienta a passar de um nível para outro.

2.2.1.5. Sugestões de melhoria

As sugestões de melhorias podem ser definidas nos termos mais alto dos níveis de maturidade e são obtidas através da verificação dos níveis de maturidades de testes. As sugestões fornecem melhorias para os processos com maturidade mais baixa.

2.2.2. Test Maturity Model Integrated – TMMI

O *Test Maturity Model Integrated* – TMMI foi desenvolvido pela TMMi Foundation. É um modelo detalhado para a melhoria do processo de teste e está posicionada como sendo complementar ao CMMI. O TMMi utiliza o conceito de níveis de maturidade para avaliação do processo e melhoria e as áreas de processo, metas e práticas são identificadas (VEENENDAAL, 2012).

O TMMI possui cinco níveis de maturidade que são: Inicial, Gerenciado, Definido, Mensurado e Otimizado. De acordo com Veenendaal (2012), cada nível tem um conjunto de áreas de processo que uma Organização precisa implementar para alcançar a maturidade a esse nível, e estão descritos na Figura 4:

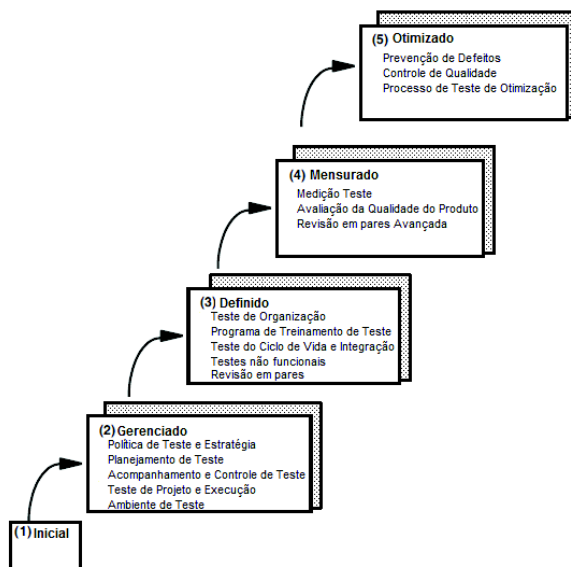


Figura 4. Níveis de maturidade TMMI

Um nível de maturidade informa o que aplicar a fim de atingir um determinado nível. Para alcançar um nível de maturidade, uma Organização deve satisfazer todos os objetivos apropriados (ambos específicos e genéricos) das áreas de processo no nível específico e também aqueles dos níveis de maturidade anteriores.

Cada nível de maturidade, exceto o nível 1, possui várias áreas de processos que indica onde deve-se focar para melhorar o processo de teste através de um conjunto de atividades e questões que precisam ser abordadas para alcançar um nível de maturidade. Cada área de processo possui objetivos específicos e genéricos.

Os cinco níveis de maturidade são descritos a seguir:

Nível 1 – Inicial: Neste nível, os testes são caóticos e não existe um processo definido. Não é identificada nenhuma área de processo.

Nível 2 – Gerenciado: No nível 2, é definido um processo com estratégia de teste, Plano de teste e casos de teste com base nos requisitos.

Nível 3 – Definido: Neste nível, o teste é integrado com o ciclo de desenvolvimento de software. O planejamento dos testes ocorre no estágio inicial do projeto, durante a fase de requisitos.

Nível 4 – Mensurado: No TMMi nível 4, os testes são completamente definidos, bem fundamentados e medidos. São considerados como parte dos testes revisões, inspeções e os produtos de software são avaliados a partir de critérios de qualidade.

Nível 5 – Otimizado: No nível cinco, testar é um processo com objetivo de prevenir defeitos, possui métodos e técnicas otimizados e estão em melhoramento contínuo.

2.2.3. Melhoria do Processo de Teste Brasileiro – MPT. Br

O MPT.Br é um modelo que trata a melhoria do processo de teste através de melhores práticas relativas às atividades desenvolvidas ao longo do ciclo de vida de teste do produto (SOFTEXRECIFE, 2011). Atualmente, o modelo MPT.Br está sendo desenvolvido e gerenciado pelas instituições SOFTEXRECIFE e RIOSOFT e tem como principal objetivo tornar-se um modelo de referência para definição, implantação e melhoria dos processos de teste.

O MPT. Br é composto por 5 níveis de maturidade. Segundo a Softexrecife (2011), cada nível possui um conjunto de áreas de processo conforme Figura 5:



Figura 5. Níveis de maturidade MPT. Br

Cada área de processo é um agrupamento de práticas relacionadas que satisfazem um objetivo e cada nível de maturidade é associado a um conjunto de práticas genéricas que devem ser aplicadas a cada área de processo que compõe o nível de maturidade desejado. Para atender a um determinado nível de maturidade, o processo de teste deve atender a todas as áreas de processo daquele nível e todos os níveis anteriores de maturidade, juntamente com as práticas genéricas.

Os cinco níveis de maturidade são descritos a seguir:

Nível 1 – Parcialmente gerenciado: Este nível contém o mínimo que uma Organização precisa para mostrar que o teste é aplicado nos projetos de forma planejada e controlada.

Nível 2 – Gerenciado: O processo de teste possui maior visibilidade e o escopo do projeto é controlado pelo processo de gestão de mudanças.

Nível 3 – Definido: São adotados processos padrões de teste e a garantia da qualidade é instituída.

Nível 4 – Prevenção de defeitos: No quarto nível, existe um processo de gestão de defeitos, onde estes, quando encontrados mais cedo no ciclo de vida, são acompanhados e ações proativas são tomadas para evitar que novos defeitos originados pelas mesmas causas raiz ocorram.

Nível 5 – Automação e Otimização: O quinto nível do modelo tem como objetivo estabelecer um processo de melhoria contínua e automação do teste.

2.3. Trabalhos Relacionados e conclusão da Fundamentação teórica

Através de pesquisas, foram encontrados alguns trabalhos relacionados a avaliação do nível de maturidade do processo de teste. Algumas propostas assemelham-se, porém são utilizados modelos de maturidade de testes diferentes para avaliar o processo de teste. Os objetivos dos trabalhos são descritos a seguir.

O trabalho de Cruz (2010), apresenta uma avaliação do processo de teste de software de uma Organização, analisando dois projetos, comparando as práticas dos projetos com as práticas do processo Gerência de Projetos de Teste de Software do nível 1 do modelo MPT.Br.

O objetivo do trabalho de Araújo (2013) foi definir um arcabouço para avaliação do nível de maturidade de processo de teste, baseado nas práticas do *Test Maturity Model Integration* - TMMI, e que fosse adequado à realidade de micros e pequenas empresas. Este arcabouço permite que as Organizações avaliem a maturidade do processo de teste sem precisar possuir um conhecimento avançado do modelo.

Embora existam modelos de maturidade de testes criados para melhorar o processo de testes das Organizações, o *Test Process Improvement* - TPI mostrou ser o modelo mais adequado para aplicar numa Organização, pois abrange todas as áreas do processo de teste. O TMMI e MPT Br., modelos citados na fundamentação teórica, estabelecem níveis de maturidade nos quais uma empresa pode alcançar, caso ela atenda aos requisitos estabelecidos para aquele nível. Estes modelos explicam quais requisitos devem ser alcançados para que a Organização conquiste um determinado nível, mas o TPI, modelo escolhido para o estudo de caso deste trabalho, sugere melhorias para alcançar e elevar os níveis de maturidade, que também podem ser verificadas nos pontos de verificação dos níveis mais altos. Os pontos de verificação e melhorias propostas são muito parecidos com os processos de testes das organizações atuais, facilitando a aplicação deste modelo.

3. Metodologia

Este capítulo abordará o método de pesquisa utilizado para execução do trabalho, assim como delineamento da pesquisa, a população alvo, técnicas de coleta e análise de dados, limitações do estudo e etapas desenvolvidas.

Neste trabalho, foi utilizada a abordagem de pesquisa qualitativa. A pesquisa qualitativa baseia-se na observação cuidadosa dos ambientes onde o sistema está sendo usado ou onde será usado, do entendimento das várias perspectivas dos usuários ou potenciais usuários do sistema. Métodos qualitativos se ocupam de variáveis que não podem ser medidas, apenas observadas (WAINER,2007).

Como método de pesquisa qualitativa, foi utilizado o estudo de caso para poder avaliar o processo de teste atual da Organização através de pontos de verificação com base em entrevistas e propor melhorias. Um estudo de caso é uma investigação empírica que investiga um fenômeno contemporâneo dentro de seu contexto da vida real, especialmente quando os limites entre o fenômeno e o contexto não estão claramente definidos (YIN, 2001).

Quanto aos objetivos, o trabalho realizado foi classificado como exploratório. As pesquisas exploratórias têm como propósito proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses. A coleta de dados pode ocorrer de diversas maneiras, mas geralmente envolve: levantamento bibliográfico; entrevistas com pessoas que tiveram experiência prática com o assunto; e análise de exemplos que estimulem a compreensão (GIL, 2010).

Para realização do trabalho, foi feito um estudo sobre conceitos importantes relacionados a teste de software e alguns modelos de maturidade de teste, detalhando o *Test Process Improvement* – TPI, modelo utilizado para realizar a avaliação do processo de testes. Com base no estudo realizado sobre o modelo de maturidade *Test Process Improvement* – TPI foi realizada uma avaliação do processo de teste da Organização. O resultado da avaliação indicou o nível de maturidade em teste de cada área chave. Foram sugeridas melhorias para elevar o nível de maturidade e melhorar o processo de teste.

As pessoas envolvidas no estudo de caso foram duas Analistas de Testes de uma equipe de desenvolvimento da Organização “ABC”, e o Coordenador de equipe. Este grupo foi escolhido, utilizando a técnica de amostragem por conveniência. Nesta técnica, os itens são escolhidos simplesmente por serem mais acessíveis ou por serem mais fáceis de serem avaliados. (GRESSLER, 2004)

A coleta de dados foi realizada através de entrevistas com profissionais da área de testes de software. Foram utilizadas questões, que são chamadas pontos de verificação, das 20 áreas chaves do *Test Process Improvement* – TPI, com o propósito de avaliar o processo de teste da Organização. O modelo de avaliação do TPI encontra-se no Anexo A. As melhorias propostas foram validadas com Analistas de testes e o Coordenador de equipe através do formulário, que encontra-se no Apêndice A, para analisar quais melhorias são adequadas para a Organização. Para a coleta de dados, a fonte de evidência observação participante também foi utilizada, pois a autora do trabalho atua como Analista de testes na Organização em estudo. A observação participante é uma modalidade na qual pode-se assumir uma variedade de funções dentro de um estudo de caso e pode, de fato, participar dos eventos que estão sendo estudados. É a capacidade de perceber a realidade do ponto de vista de alguém de "dentro" do estudo de caso, e não de um ponto de vista externo (YIN, 2001).

Foram analisados os artefatos no repositório do projeto como: casos de teste, documentos, documentação de análise e ferramentas utilizadas pela equipe de teste para

auxiliar na avaliação e proposta de melhorias. A análise documental consiste em uma série de operações que visam estudar e analisar um ou vários documentos para descobrir as circunstâncias sociais e econômicas com os quais podem estar relacionados (RICHARDSON, 1999).

A análise de dados da avaliação foi realizada com base no resultado dos pontos de verificação. Cada área chave do *Test Process Improvement* – TPI possui níveis de A até D com pontos de verificação (*checkpoints*). Para alcançar um determinado nível, todos os pontos de verificação devem ser atendidos, juntamente com os pontos de verificação do nível anterior se houver. Para alcançar um nível de maturidade, todos os pontos de verificação daquele nível devem ser atingidos completamente. Foi mostrado graficamente o percentual total alcançado por área chave e o percentual alcançado por nível da área chave. Cada ponto de verificação (*checkpoint*) pode ser avaliado como Sim, Não ou Parcialmente. No Quadro 2 é apresentada a descrição das respostas e o seu percentual utilizados na avaliação:

Quadro 2. Descrição das respostas e seu percentual

Resposta	Percentual	Descrição
Sim	100%	O ponto de verificação foi atingido. Os procedimentos são utilizados
Parcialmente	50%	O ponto de verificação foi parcialmente atingido. Nem todos os procedimentos são utilizados.
Não	0%	O ponto de verificação não foi atingido. Procedimentos inexistentes.

Para mensurar o grau de concordância dos participantes que responderam os questionários das melhorias foi utilizada a escala Likert, com graduações de 1 a 5. A escala Likert é uma escala de mensuração com cinco categorias de respostas, variando de "discordo totalmente" a "concordo totalmente", que exige que os participantes indiquem um grau de concordância ou de discordância com cada uma das várias afirmações relacionadas aos objetos de estímulo (MALHOTRA, 2012).

No estudo de caso não foram implementadas as melhorias para elevar o nível de maturidade devido à limitação de tempo para a realização deste trabalho. Foi avaliado o processo de teste somente de uma equipe de desenvolvimento da Organização responsável por alguns módulos do sistema financeiro, que será detalhado no estudo de caso do capítulo 4. Esta equipe é composta por Analistas de sistemas, Desenvolvedores e Analistas de testes. As melhorias propostas, se adequadas, poderão ser implementadas futuramente para todas as equipes da Organização para melhorar o processo de teste.

O trabalho foi realizado com as seguintes etapas: construção da fundamentação teórica do trabalho; realização de um estudo sobre a avaliação do processo de teste utilizando o modelo *Test Process Improvement* – TPI; definição de como aplicar o estudo de caso e os participantes; realização do estudo de caso executando a avaliação do processo teste de software utilizando o modelo *Test Process Improvement* – TPI e através da análise dos resultados propor e validar as melhorias para o processo de teste da Organização.

4. Estudo de caso

4.1. Organização em estudo

A empresa “ABC”, na qual a avaliação do TPI foi aplicada, é especializada em soluções tecnológicas para o segmento financeiro e atende em diversas regiões do Brasil. A empresa tem em torno de 180 profissionais entre eles funcionários, aprendizes e estagiários. A área de Suporte da empresa presta assistência aos clientes a fim de solucionar problemas técnicos do sistema antes de repassar para a equipe de Fábrica de software. A Fábrica de software é responsável pela manutenção do sistema e é composta por Analistas de sistemas, Desenvolvedores e Analistas de teste. O principal sistema utilizado é um sistema para instituições financeiras, cooperativas de crédito e bancos. O sistema possui módulos integrados como cadastro, empréstimo, aplicação, capital, e comunicação entre outros.

A Organização utiliza a metodologia *Scrum* no seu processo de desenvolvimento de software. O *Scrum* enfatiza o uso de um conjunto de padrões de software que se mostrou efetivo para projetos com cronogramas apertados, requisitos mutáveis e aspectos críticos de negócio. Cada padrão de processo define um conjunto de tarefas de desenvolvimento e permite à equipe *Scrum* construir um processo que se adapta às necessidades do projeto (PRESSMAN, 2011).

No processo de desenvolvimento do sistema, o Analista de sistemas recebe uma demanda que pode ser uma correção ou melhoria do sistema. A demanda é analisada e o Analista de sistemas verifica se o documento de especificação funcional de requisitos precisa ser criado ou somente alterado para descrever as modificações que deverão ser efetuadas. A equipe de testes recebe o documento de especificação e inicia a escrita dos casos de teste em paralelo com a equipe de desenvolvimento que começa a implementação do sistema. Após a implementação do sistema, o ambiente de testes é preparado e inicia-se a execução dos testes para validar se os resultados obtidos estão em conformidade com a solução técnica especificada. Quando encontrado um erro durante a execução dos testes, as inconformidades são registradas para que o Desenvolvedor o corrija. Quando não houver mais inconformidades no sistema, a equipe de teste encerra as atividades de teste, comunicando o término da validação.

4.2. Execução e análise da Avaliação

A avaliação do processo de teste de software da Organização “ABC” foi realizada com duas Analistas de testes através de uma reunião marcada no dia 02 de julho de 2015, juntamente com a participação da autora do trabalho. A avaliação foi conduzida pela autora do trabalho através de entrevistas e foi utilizado o modelo *Test Process Improvement* – TPI, com as 20 áreas chaves do processo de teste e seus pontos de verificação, no qual foi preenchido com as respostas das Analistas de testes. A atividade da avaliação definiu o atual processo de testes da Organização. Os resultados encontram-se descritos nos itens do Anexo A. Através dos dados coletados chegaram-se as seguintes conclusões:

- A área chave Estratégia de teste não alcançou o nível A, pois o ponto de verificação sobre “uma estratégia para decidir entre testar somente os problemas resolvidos ou retestar tudo” não foi atingido. Geralmente não há uma estratégia e a maioria das

demandas de erro são urgentes, focando somente no teste do erro em específico que foi corrigido. Portanto, a área chave **Estratégia de teste** encontra-se no **nível inicial < A**. Os riscos do sistema são considerados pela experiência que o testador tem na empresa e conhecimento do sistema e não é utilizada uma documentação de análise de riscos.

- No Modelo de ciclo de vida não foi alcançado o nível A pela ausência de um Plano de teste. Na Organização não é elaborado um Plano de teste devido à existência de muitas demandas de erros e emergenciais para corrigir pontos específicos do sistema. Apesar de não ter um Plano de teste, o planejamento das atividades de teste é realizado informando recursos, horas e o que deve ser entregue. A área chave **Modelo de ciclo de vida** encontra-se no **nível inicial < A**.

- A área chave **Momento do envolvimento** conquistou o **nível A** de maturidade, pois a atividade de teste começa após a conclusão da documentação que serve de base para o teste. No *Test Process Improvement* – TPI, para evoluir entre as escalas da matriz de maturidade é necessário que sejam respeitadas as dependências que existem entre os níveis de maturidade e as áreas chaves. O nível A, da área chave Momento do envolvimento é dependente no nível A, da área chave Modelo de ciclo de vida. O nível A da área chave Modelo de ciclo de vida não foi alcançado, mas considerando que a atividade de teste começa mais cedo, após a conclusão da documentação e antes que o sistema esteja desenvolvido, a área chave Momento do envolvimento ficará no nível A.

- O Planejamento e estimativa das atividades de teste são definidos na “reunião de *Sprint*”, onde todas as tarefas das demandas de erro e implementações são exibidas com o planejamento das atividades de execução e caso de teste e estimativa de quantas horas serão necessárias para a realização das tarefas. No decorrer da *Sprint* ajuste de horas poderão feitos se necessário. Sendo assim, a área chave Planejamento e estimativa alcançou o nível A de maturidade. Este nível é dependente do nível A da área chave Modelo de ciclo de vida que não foi alcançado, mas como as atividades do processo de teste estão bem definidos no modelo de ciclo de vida, a área chave **Planejamento e estimativa** continua no **nível A**.

- A equipe de testes possui uma técnica de especificação de teste documentada, porém não é seguida por todos. Não existe uma técnica padronizada para a escrita de casos de teste, onde todos os testadores deveriam seguir para que os casos de teste fossem escritos de forma completa e de acordo com a documentação. Portanto, a área chave **Técnicas de especificação de testes** atingiu o **nível A** de maturidade.

- Técnicas estáticas não são utilizadas para avaliar a documentação. O Analista de testes ao criar os casos de teste, caso haja alguma inconformidade, notifica o Analista de sistemas para que a documentação seja corrigida. A área chave **Técnicas estáticas** de teste encontra-se no **nível inicial < A**.

- A equipe de testes não utiliza métricas para avaliar a qualidade do sistema, somente o Coordenador de equipe utiliza métricas como: “Quantidade de recursos e horas utilizadas por recurso” e horas “previstas X horas realizadas por atividade”. A área chave **Métricas** não atingiu todos os pontos de verificação no nível A, portanto está no **nível inicial < A**.

- Ferramentas para auxiliar o trabalho dos testadores são utilizadas na realização das tarefas de escrita e execução de teste, gerenciamento dos defeitos, controle e

apontamento de horas, captura de imagens para as evidências de teste. Algumas destas ferramentas são livres, por este motivo os testadores não tem uma visão do custo das ferramentas. A área chave **Ferramentas de teste** encontra-se no **nível A**.

- Os testes são realizados no ambiente de teste igual ao de produção, isto é, podem ser reproduzidos os erros utilizando uma base de dados do dia em que foi detectado o problema em produção. Cada testador possui seu próprio ambiente de teste e as solicitações de carga de base são realizadas em tempo hábil. A cópia de versões do sistema no ambiente de desenvolvimento é realizada facilmente para a execução dos testes. Estes procedimentos são realizados sem autorização do Gerente de teste ou Coordenador de equipe, pois a equipe de teste possui autonomia para realizar alterações no ambiente de teste, tornando o trabalho mais eficiente e agilizado. Por este motivo a área chave **Ambiente de testes** encontra-se no **nível inicial < A**.

- O local de trabalho dos testadores é considerado adequado e possui toda infraestrutura necessária para realizar as atividades. Os testadores ficam próximos dos desenvolvedores e da equipe do projeto para ter mais produtividade e trabalhar em conjunto. A área chave **Local de trabalho** atingiu o **nível A**, o único da área chave.

- A área chave **Motivação e comprometimento** foi quase totalmente alcançada. A equipe de teste tem um bom relacionamento com os integrantes do projeto e comprometimento em entregar um sistema de qualidade. A equipe de teste considera que a área de teste não tem um status alto, não é vista como uma "promoção" dentro da Organização. A área chave **Motivação e comprometimento** alcançou o nível B de maturidade, mas este nível é dependente das seguintes áreas chaves: nível A do Modelo de ciclo de vida, nível B de Relatórios e nível B de Gerenciamento do processo de teste. Os níveis destas áreas chaves não foram atingidos, como exemplo: relatórios com o progresso das atividades e status dos defeitos não são elaborados, portanto a área chave **Motivação e comprometimento** está no **nível A**.

- A Organização não possui treinamentos sobre o negócio e qualificação na área de testes de software. O conhecimento adquirido pelos Analistas de testes é pela experiência no trabalho e por sanar dúvidas com as pessoas envolvidas no projeto. No processo de teste da Organização não é realizado o teste de aceitação com o usuário final, pois a maioria das manutenções são correções de erros específicos no sistema e muitas vezes o tempo necessário para entrega da versão para a produção é muito curto. A área chave **Atividades e treinamento de testes** encontra-se no **nível inicial < A** por não alcançar todos os pontos de verificação do nível A.

- Para cada projeto não é formulada e nem seguida nenhuma metodologia para realizar as atividades de teste. A área chave **Escopo da metodologia** encontra-se no **nível inicial < A**.

- A equipe do projeto tem reuniões diárias para comunicar o andamento do projeto, chamada "*Daily*". Muitas vezes uma reunião para comunicar o status do projeto não é realizada por falta de tempo, isto pode contribuir para impactos negativos. Para tratar de assuntos relacionados a atividades de teste e processo, a equipe de teste não tem reuniões. Quando há problemas no projeto, uma comunicação informal entre o testador e o Coordenador de equipe é realizada. Todos os testadores da Organização tem uma reunião de qualidade para assuntos relacionados à área de teste, mas não são frequentes.

A área chave **Comunicação** está no **nível inicial < A** por não atingir todos os pontos de verificação do nível A.

- No projeto não são enviados relatórios para comunicar a situação atual do projeto para a equipe com quantidade de defeitos abertos, resolvidos ou impeditivos. A área chave **Relatórios** está no **nível inicial < A** por não alcançar nenhum dos pontos de verificação.

- O gerenciamento dos defeitos é realizado através da ferramenta Mantis, onde é possível reportar o defeito para o desenvolvedor informando o título, a descrição do problema e os passos para reproduzi-lo, o resultado esperado, status, severidade entre outros. A responsabilidade por garantir que a administração dos defeitos é realizada corretamente e de forma consistente é do próprio testador. A área chave **Gerenciamento dos defeitos** alcançou o **nível A** de maturidade.

- Os produtos de teste como casos de teste, documentação e versões do sistema disponibilizadas para teste tem um bom gerenciamento. Os casos de teste são reutilizáveis, podendo ser criados novas versões e é possível saber quem foi o último usuário a alterar o caso de teste. No caso de teste, é informada qual versão do sistema será executado o teste e o link da documentação utilizada para criação do caso de teste. É possível verificar o histórico do todo o caso de teste e suas versões anteriores. A documentação do sistema possui um controle de versões, com data da última atualização e o usuário. Quando alguma alteração é feita do documento, o Analista de sistemas notifica o testador para que o caso de teste seja modificado. Os objetos das versões do sistema também possuem um controle de versão para gerenciamento das alterações realizadas. Quando uma nova versão do sistema é criada, o testador é notificado pela equipe de desenvolvimento para que possa copiar a versão do ambiente de desenvolvimento para o ambiente de testes e iniciar a execução dos testes. A área chave **Gerenciamento dos produtos de teste** alcançou todos os níveis, mas como o nível C é dependente do nível B de Técnicas de especificação de testes, e este não foi atingido por não ter uma técnica formal para criar os casos de testes, a área chave **Gerenciamento dos produtos de teste** ficará no **nível B**, pois para reutilizar os casos de teste, uma técnica formal deve ser utilizada e reconhecida por todos.

- Há um planejamento das atividades de teste, tempo, recursos e produtos a serem entregues, mas não é criado um Plano de teste, assim como não é seguida uma metodologia no processo de teste. A área chave **Gerenciamento do processo de teste** encontra-se no **nível inicial < A** por não contemplar o nível A.

- A Avaliação de produtos intermediários, tais como requisitos funcionais, utilizando técnicas de avaliação não é realizada. Na escrita dos casos de testes, caso seja encontrado alguma inconformidade, é sinalizado ao Analista de sistemas para que seja feita a correção. A área chave **Avaliação** encontra-se no **nível inicial < A**.

- Os desenvolvedores não utilizam nenhuma técnica ou estratégia para realizar o teste unitário. Não há nenhum planejamento de testes, apenas validam o que foi corrigido ou implementado. Muitas vezes o teste unitário não é realizado e o sistema é entregue para o testador validar, apresentando muitos defeitos e comprometendo o trabalho de toda a equipe. A área chave **Testes de baixo nível** encontra-se no **nível inicial < A**.

O relacionamento entre as áreas chaves e níveis da Organização são apresentados no Quadro 3 baseado em Koomen (1999), e indicam que o processo de teste da

Organização não está no nível Controlado, pois muitas áreas chaves não conseguiram alcançar o nível A ou B. A matriz de maturidade mostra o nível que cada área chave alcançou na Organização e as dependências que existem entre as áreas chaves.

Quadro 3. Matriz de maturidade e dependência de testes

Escalas	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Áreas chave	Inicial	Controlado				Eficiente				Otimizado				
1 Estratégia de teste		A (5A, 11A)					B (2A,5B,11B, 14B,18B)				C (3C,19B) ou 20C		D (3C, 19B, 20C)	
2 Modelo de ciclo de vida		A (11A)				B (6A, 17A)								
3 Momento do envolvimento			A (2A)				B (2B)				C		D (11C)	
4 Planejamento e estimativa				A (2A)							B (7B, 15B)			
5 Técnicas de especificação de testes		A			B (12A, 17A)									
6 Técnicas estáticas de teste					A		B							
7 Métricas						A (11B, 15B, 16A, 18B)			B (15C, 16B)			C (13B, 14C, 18C)		D
8 Ferramentas de teste					A			B (5B, 12A)			C			
9 Ambiente de testes				A (12A)				B(1B)						C
10 Local de trabalho				A										
11 Motivação e comprometimento		A				B (2A, 15B, 16A, 18B)							C (1C, 3C, 8B, 15C)	
12 Atividade e treinamento de testes				A			B				C (13A)			
13 Escopo da metodologia					A (2A, 5 B, 16A, 17A, 18B)						B			C (11B, 18C)
14 Comunicação			A		B (2A, 15B, 16A)								C (13B)	
15 Relatórios		A			B (2A, 16A, 18B)		C (1A, 5B, 7A, 16B)						D (1C, 11C)	
16 Gerenciamento dos defeitos		A				B		C						
17 Gerenciamento dos produtos de teste			A			B				C (5B)				D
18 Gerenciamento do processo de teste		A		B								C (13B)		
19 Avaliação							A			B				
20 Teste de baixo nível					A		B		C					

Através da análise realizada conclui-se que a Organização atingiu um percentual de 27% do total da matriz de maturidade de testes e das dependências entre as áreas chaves e os níveis de testes. A Figura 6 mostra a média por área chave e o percentual total dos pontos de verificação alcançados, levando em consideração o nível que cada área chave alcançou no modelo *Test Process Improvement – TPI*.

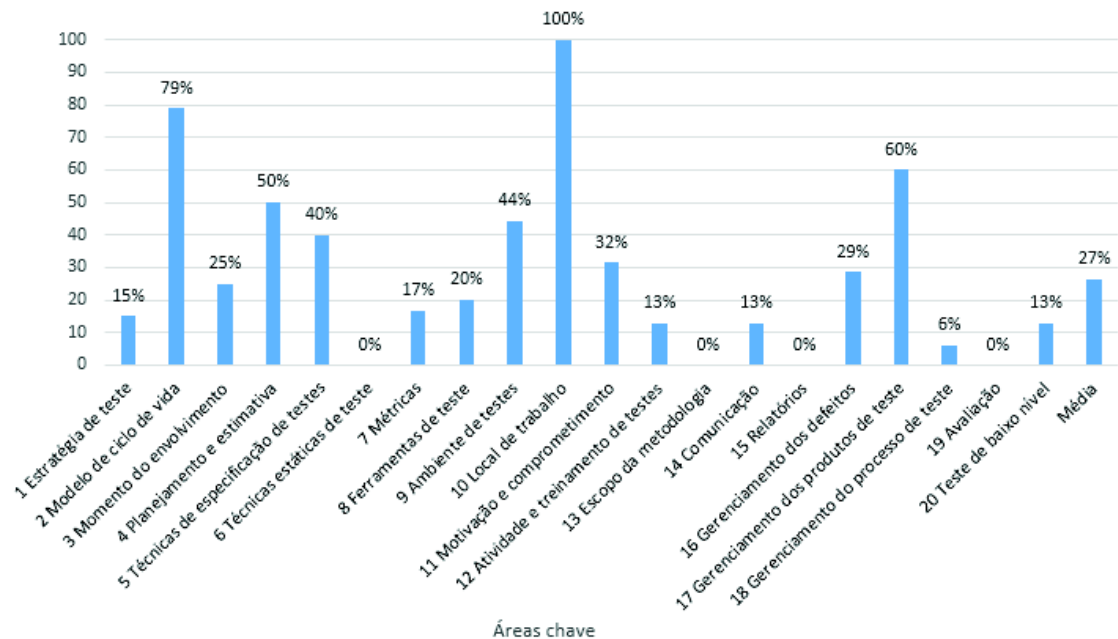


Figura 6. Média por área chave

O percentual atingido pela Organização dos níveis A e B é apresentado na Figura 7. As áreas chaves que alcançaram um percentual abaixo de 100% para o nível A são consideradas como nível inicial < A. Para ter atingido o nível A de maturidade, o nível deve ter seu percentual em 100%.

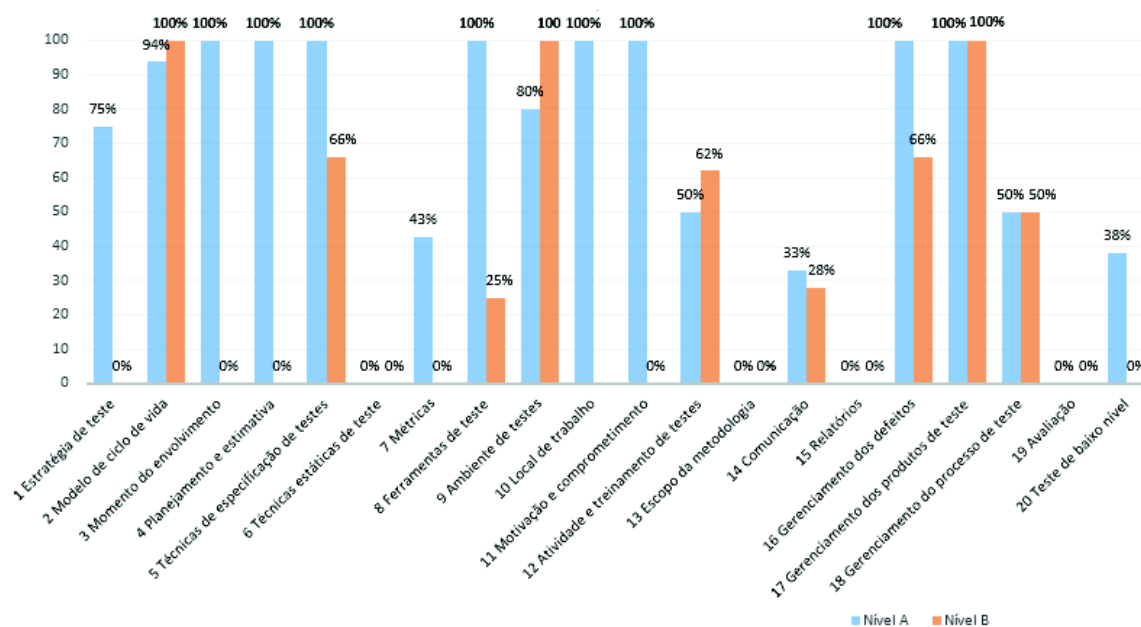


Figura 7. Média dos níveis A e B

4.3. Sugestões de melhorias para o Processo de teste

Estratégia de testes:

A área chave Estratégia de testes não alcançou o nível A de maturidade. Como melhoria, é sugerido o teste de regressão para a garantia da qualidade do sistema e para evitar que uma funcionalidade não deixe de funcionar após a implementação ou manutenção do sistema. O teste de regressão retesta segmentos já testados após a implementação de uma mudança em outra parte do software. Visa garantir a integridade do software depois da realização de novos testes (BASTOS, 2007). Para retestes, avaliar se é necessário testar somente o problema resolvido ou retestar tudo.

Ter uma estratégia de coordenação entre os diferentes níveis de teste como testes de sistema, unitário e de aceitação. Esta melhoria é importante para definir qual a cobertura de testes e os riscos envolvidos. A comunicação entre o testador e o desenvolvedor poderá ajudar na escolha de quais testes unitários são importantes realizar antes de entregar o sistema para a equipe de teste validar, evitando defeitos e aumentando a qualidade do sistema. Os testadores podem auxiliar no teste de aceitação, indicando para o cliente quais testes são mais importantes de serem executados com a ajuda dos casos de teste.

Modelo de ciclo de vida

A área chave Modelo de ciclo de vida não alcançou o nível A. Sugere-se, como melhoria para projetos maiores e novas implementações do sistema, a utilização de um Plano de teste no planejamento de testes para ter o projeto documentado, definir a cobertura de testes e ter informações sobre o sistema que será testado. O Plano de testes

é o documento que contém o projeto ou desenho lógico do processo de testes e está alinhado à Estratégia de Testes correspondente e nele começam a ser delineados os casos de teste. Este documento define os objetivos gerais esperados e as expectativas do projeto de testes (RIOS,2013).

Momento do envolvimento

A área chave Momento do envolvimento conquistou o nível A de maturidade, e como melhoria para evoluir de nível, sugere-se envolver os testadores logo no início do projeto ou especificação de requisitos para diminuir o custo do projeto final, evitar defeitos futuros, garantir que os requisitos mencionados estejam corretos e sem ambiguidades e, ainda, indicar requisitos que seriam importantes para a implementação do sistema. A condição mais importante é que o testador tenha conhecimentos e experiência suficiente para auxiliar o Analista de sistemas e Analista de negócios na construção dos requisitos do sistema.

Planejamento e estimativa

As melhorias indicadas para a área chave Planejamento e estimativa também são relacionadas com as áreas chave Métricas e Relatórios. Implementando as melhorias indicadas para estas áreas, a área chave Planejamento e estimativa poderá subir o nível de maturidade para B.

Técnicas de especificação de teste

A área chave Técnicas de especificação de teste atingiu o nível A e para atingir o nível B, a melhoria proposta é utilizar uma técnica de especificação de casos de teste para elaborar casos de teste padronizados e com instruções detalhadas, de modo que o testador que irá executar o teste tenha informações suficientes para conduzir de forma correta e sem dúvidas o passo a passo do caso de teste descrito, contendo todas as regras da documentação do sistema. Documentos padronizados facilitam a comunicação entre as partes envolvidas, pois definem uma forma de comunicação comum. A técnica deverá ser documentada e utilizada por todos os testadores da empresa.

Técnicas estáticas de teste

A área chave Técnicas estáticas de teste não alcançou o nível A. A melhoria proposta pelo TPI é utilizar uma técnica baseada em *Checklists*, onde é formada uma equipe de inspeção, podendo ser os próprios testadores, utilizando uma lista com os itens a serem verificados e o que deve ser entendido como defeito. Utilizando uma técnica estática a documentação do sistema irá atender as especificações do cliente e estará dentro dos padrões especificados. A detecção de problemas na documentação no início do projeto evita o alto custo de retrabalho e manutenções futuras.

Métricas

A área chave Métricas não atingiu o nível A de maturidade. As métricas podem ser utilizadas para avaliar a produtividade, medindo o processo; e para avaliar a qualidade, medindo o produto. As métricas de produto como melhoria, são consideradas mais adequadas para se aplicar na Organização. A equipe de teste que vai trabalhar com métricas deve ter um treinamento para selecionar um conjunto de métricas coerentes e mais simples no início, que agreguem valor ao processo e que sejam importantes para as atividades de testes.

Alguns exemplos de métricas são: analisar o índice de severidade dos defeitos (baixo, normal, alto, crítico) para verificar quais defeitos são mais graves; utilizar métricas de cobertura dos testes para verificar se todas as funcionalidades do sistema estão sendo testadas, como por exemplo, a quantidade de casos de testes planejados versus executados; e métricas do progresso do teste, para verificar quantos e quais testes foram executados e o status de cada teste (concluído / não concluído).

As métricas de produto podem ser úteis para avaliar a quantidade de defeitos que ocorreram entre as diferentes versões do sistema, quais as possíveis causas e ajudar na tomada de decisões para melhorar a qualidade do sistema. As métricas podem ser utilizadas para mostrar para a equipe do projeto como está a qualidade do sistema testado e devem ser exibidas num relatório.

Ferramentas de teste:

O custo da ferramenta não é um fator interessante para os testadores, por este motivo a área chave Ferramentas de teste conquistou somente o nível A. A ferramenta utilizada para auxiliar o processo de gerenciamento do teste é o Testlink, mas não atende as expectativas dos testadores causando muitas vezes transtornos e atrasos na construção dos casos de teste. Visando maior agilidade nas atividades do processo de testes como melhoria, a utilização de uma nova ferramenta poderia contribuir para consideráveis ganhos de tempo, produtividade, confiabilidade, qualidade na construção e execução dos casos de teste e em todo ciclo de vida do teste.

Ambiente de teste

Os testadores tem autonomia para atualizar o ambiente de testes com novas versões e solicitar carga de base para realizar os testes sem a necessidade de autorização do Gerente de testes ou Coordenador de equipe, por este motivo a área chave Ambiente de teste não conquistou todos os níveis. Sem a autorização para realizar os procedimentos, os testadores tem mais agilidade para executar as tarefas com mais rapidez e autoconfiança, evitando atrasos pela pendência ou falta de uma autorização, por esta razão não será indicado melhorias.

Motivação e comprometimento

As melhorias para a área chave Motivação e comprometimento são indicadas nas áreas chaves dependentes: Modelo de ciclo de vida, Relatórios e Gerenciamento do processo de teste para que esta possa alcançar o nível B de maturidade. Os testadores consideram que a área de testes possui salários mais baixos que as demais áreas, sendo assim como melhoria, levar para o setor de Recursos Humanos a importância da área de testes dentro da Organização informando as vantagens e o papel importante que tem para que o produto seja entregue com qualidade, pois os testadores são os olhos do cliente. Informar que através da avaliação do processo de teste com os profissionais da área, esta questão não foi alcançada positivamente.

Atividades e treinamento de teste

A área chave Atividades e treinamento de teste não alcançou o nível A, como melhoria, propõem-se treinamentos para os testadores na busca de qualificação na área de testes de software, sobre o negócio e o sistema, que são importantes para preparação técnica, desenvolvimento de novas competências que auxiliem nas atividades da forma

mais eficaz possível. Treinamentos sobre o negócio da empresa e como o sistema funciona, são os mais importantes e deveriam ser considerados como requisitos para poder exercer as funções do trabalho e auxiliar na execução do teste tornando assim um sistema de qualidade no final.

Para projetos mais críticos e com tempo de duração maiores, o teste de aceitação como melhoria, seria de grande importância para validar com o usuário se os requisitos estão de acordo com o que foi solicitado e para a aprovação da solução entregue. Disponibilizar para o cliente o conhecimento necessário para realizar os testes de aceitação com auxílio dos Analistas de testes para conduzir os testes no sistema. Com o teste de aceitação o retrabalho, perdas, prejuízos e defeitos em produção seriam evitados. O teste de aceitação é a última ação de teste antes da implantação do software, sendo sua execução de responsabilidade do cliente. O objetivo do teste de aceitação é verificar se o software está pronto e pode ser usado por usuários finais para executar as funções e tarefas para os quais foi construído (BASTOS,2007).

Para garantir que os padrões de qualidade estão sendo cumpridos e seguidos de acordo com os padrões e metodologia é importante ter um Analista de qualidade para auditar os procedimentos, atividades e o processo de testes.

Escopo da metodologia

A área chave Escopo da metodologia não alcançou o nível A, como melhoria proposta sugere-se definir uma metodologia de teste com o propósito de contribuir para o alcance da qualidade do sistema e ter um processo focado na melhoria do ciclo de desenvolvimento. Uma metodologia de teste é necessária pelo fato de existirem fatores que afetam a realização eficiente de um processo de teste como: não testar todos os requisitos, processos formais inexistentes ou mal definidos, pouca importância atribuída ao processo de teste, equipe não qualificada e defeitos só detectados em produção. Estes fatores dificultam o processo de aquisição da qualidade do sistema e podem aumentar os custos e retrabalho.

A metodologia deve conter as etapas importantes para a realização das atividades de teste como: validação de requisitos, critério de escolha do que será testado levando em consideração uma análise de riscos, formalizar um Plano de teste, especificar o projeto de teste, especificar os casos de teste padronizados, executar os testes, relatar os incidentes e avaliar e documentar os resultados através de relatórios.

Comunicação

A área chave Comunicação não conseguiu atingir o nível A. Como melhoria, realizar reuniões da equipe de teste com o Coordenador da equipe para relatar o andamento das atividades, os problemas enfrentados no projeto, os impedimentos, o que precisa ser melhorado no processo de teste e reportar os defeitos mais críticos. Quando o projeto tem uma boa comunicação entre os envolvidos, um plano de ação pode ser iniciado para solucionar os problemas, melhorias poderão ser implementadas e análise de risco. Documentar os pontos fracos e fortes da reunião e as ações de melhoria.

Relatórios

A área chave Relatórios não alcançou o nível A de maturidade. Apresenta-se como sugestão enviar relatórios para o Coordenador de equipe e equipe de

desenvolvimento, comunicando a situação atual do projeto. No relatório devem ser informados os defeitos em aberto organizados por severidade e os impedimentos para realização das tarefas ou correção dos defeitos. Os relatórios dão uma visibilidade do andamento e progresso do projeto, os problemas que estão sendo enfrentados para a finalização e questões sobre o tempo disponível para a entrega do projeto.

Gerenciamento dos defeitos

A área chave Gerenciamento de defeitos alcançou o nível A de maturidade. Para aumentar o nível de maturidade sugere-se como melhoria, um acompanhamento do projeto e qualidade do sistema pelo Coordenador da equipe, através da ferramenta Mantis utilizada pela equipe de testes, para verificar a quantidade de defeitos em aberto, resolvidos e não resolvidos.

Gerenciamento dos produtos de teste

A área chave Gerenciamento dos produtos de teste não alcançou todos os níveis por ser dependente da área chave Técnicas de especificação de testes. As melhorias da área chave Técnicas de especificação de testes também são indicadas para Gerenciamento dos produtos de teste.

Gerenciamento do processo de teste

A área chave Gerenciamento do processo de teste não atingiu o nível A. As melhorias mencionadas para as áreas chave Modelo de ciclo de vida e Escopo da metodologia como criar um Plano de teste e seguir uma metodologia, devem ser utilizadas para avançar o nível de maturidade da área chave de Gerenciamento do processo de teste.

Avaliação

A área chave Avaliação não alcançou o nível A. A avaliação tem como objetivo avaliar produtos intermediários, tais como requisitos funcionais, a fim de encontrar defeitos nas fases iniciais reduzindo o custo de retrabalho. As avaliações verificam produtos intermediários e não como o teste de software que validam produtos finais. Segundo Rios (2013), as técnicas de avaliação existentes são:

Walk-throughs: reuniões informais para avaliação dos produtos. Pouca ou nenhuma preparação é usualmente requerida.

Revisões técnicas ou inspeções: são práticas nas quais um grupo de pessoas, por meio de procedimentos formais, se dedica a descobrir defeitos em produtos (documentos ou códigos). O processo de avaliação é de grande importância para a melhoria dos produtos desenvolvidos reduzindo prazos, custos de desenvolvimento e redução de retrabalho.

Teste de baixo nível

A área chave Teste de baixo Nível não atingiu o nível A. Conscientizar os desenvolvedores sobre a importância dos testes unitários é um dos fatores mais importantes para garantir a qualidade do software e serve como proposta de melhoria. Com o teste unitário, o desenvolvedor avalia a estrutura interna do sistema, mas também deve ser considerado avaliar os requisitos funcionais e usabilidade. Muitos desenvolvedores acreditam que testar toma muito tempo e acabam entregando o sistema para a equipe de teste validar sem ter realizado o teste unitário. Um fator que pode

minimizar este pensamento é relacionar o custo gasto quando um problema é encontrado meses depois no sistema e o desenvolvedor tem que focar seu trabalho na descoberta deste erro para depois pensar em uma solução. O teste unitário deve ser considerado como um investimento e como um retorno positivo e rápido aumentando a confiança na equipe.

Se o sistema for bem testado pelo desenvolvedor antes de passar pela equipe de testes, torna o trabalho mais simples e motivador e a equipe produzirá mais e com melhor qualidade. Não é necessário os desenvolvedores utilizarem alguma técnica para realizar os testes, para a equipe de teste o importante é receber o sistema em boas condições para poder realizar o teste funcional e ter tempo para explorar outras partes do sistema a fim de encontrar inconformidades e entregar um sistema com maior qualidade.

4.4. Validação das melhorias do Processo de teste

A validação das melhorias do processo de teste foi realizada com duas Analistas de teste, que participaram da avaliação, e o Coordenador de equipe. Foi utilizado o formulário que encontra-se no Apêndice A, contendo as melhorias de cada área chave que não atingiram um certo nível de maturidade e precisam melhorar seu processo de teste. Para validação das melhorias foi disponibilizado um documento de apoio com informações sobre a análise da avaliação do processo de teste e o que são as áreas chaves do TPI para auxiliar no caso de dúvidas sobre o porque aplicar tal melhoria.

O formulário e o documento foram enviados para o e-mail dos participantes no dia 10 de agosto de 2015, para que as perguntas sobre as melhorias fossem respondidas individualmente com tempo previsto de retorno de 2 dias. Após o preenchimento do formulário, os participantes enviaram o e-mail com as respostas sobre as melhorias propostas para que fossem analisadas.

Com base nas respostas dos formulários, as melhorias propostas foram validadas e obtiveram as seguintes justificativas para serem ou não implementadas:

- A melhoria sobre realizar testes de regressão foi aceita por todos os participantes. A melhoria proposta tem como objetivo realizar testes de regressão quando surgir uma demanda de erro ou emergencial que precisam entrar em produção e que tenham riscos para a Organização caso ocorra algum erro em outra parte do sistema.
- A melhoria ter uma estratégia de coordenação entre testes de sistema e unitário foi aceita parcialmente por um participante, pois seria necessário ter tempo disponível e um planejamento para ser realizada uma estratégia. Os demais concordaram totalmente em aplicar esta melhoria na Organização.
- A melhoria sobre ter testes de aceitação e auxílio aos cliente na sua execução não foi aceita por um participante, pois não é realizado teste de aceitação na versão liberada. A versão é utilizada em produção por algumas cooperativas pilotos e caso haja algum erro é informado para que seja corrigido. Já outros participantes acreditam que o teste de aceitação poderia ser executado, pois novos projetos e implementações devem ter cuidados redobrados, ou até mesmo para sanar dúvidas dos clientes de como utilizar o sistema antes de colocar em produção.
- A melhoria envolver o Analista de testes com experiência logo no início do projeto, foi aceita pelos Analistas de testes, mas o Coordenador de equipe discordou pois na fase de

concepção do projeto, seja da parte de negócios, quanto de sistemas, os requisitos ainda estão em estudo, e por esta razão ainda não estão totalmente prontos para serem validados pela equipe de testes.

- Utilizar uma técnica de especificação de casos de teste formal foi completamente aceita pelos participantes, apesar de existir uma documentação informal de como criar os casos de testes, não é utilizada por todos da empresa.

- Todos os participantes concordam em utilizar métricas de produto, uma sugestão para utilizar as métricas seria a cada fechamento de *Sprint*, para que os dados coletados sejam utilizados como base na próxima *Sprint* que iniciar.

- A utilização de uma nova ferramenta de testes substituindo o Testlink foi totalmente aceita pelos Analistas de teste. O Coordenador de equipe concorda parcialmente e acredita que é preciso de mais informações sobre as funcionalidades da nova ferramenta em comparação com a atual para verificar se é mesmo necessário a troca de ferramenta.

- A melhoria de levar para o setor de Recursos Humanos a importância da área de testes dentro da Organização foi aceita pelos Analistas de teste, pois acreditam que a qualidade ainda precisa ser mais trabalhada para mostrar que agrega valores e não subtrai; ou somente aponta erros, e serve para que o cliente fique mais satisfeito, assim beneficiando toda empresa. O Coordenador de equipe discorda desta melhoria, pois dentro do processo de desenvolvimento de sistemas da empresa as atividades de testes são tão importantes quanto às demais e todas as atividades de melhoria contínua dos processos incluem a área de testes, para que haja oportunidade de trazer e debater as ações de aperfeiçoamento nessa área. Também ressaltou que o Plano de Cargos e Salários contém os mesmos níveis de cargos de todas as áreas da empresa, incluindo os profissionais de teste de software.

- Todos os participantes concordam com a melhoria de realizar treinamentos para os testadores para qualificação na área de testes e negócio. O Coordenador justificou que esta questão é indicada no Plano de Desenvolvimento Individual e encaminhadas à área de Gestão de Pessoas.

- A melhoria que diz respeito a realizar reuniões da equipe de teste, foi aceita por todos os participantes. O Coordenador de equipe justificou informando que estas reuniões podem acontecer juntamente com as reuniões atuais do processo de desenvolvimento; nas reuniões diárias que ocorrem para acompanhamento e relato dos impedimentos e problemas enfrentados, e na reunião de revisão e retrospectiva de *Sprint*, realizadas para avaliar o desempenho da equipe naquela *Sprint*, onde são discutidas as melhorias e ações para a próxima.

- Todos os participantes concordaram com as seguintes melhorias: enviar relatórios sobre o projeto para os envolvidos no projeto e Coordenador; ter um acompanhamento do projeto pelo Coordenador de equipe sobre os defeitos através da ferramenta Mantis; ter um Analista de qualidade para auditar o processo de teste; utilizar uma técnica estática de teste para verificar possíveis erros na documentação do sistema; elaborar um Plano de teste e definir uma metodologia de testes.

- Utilizar técnicas de avaliação para avaliar produtos intermediários como melhoria foi aceita parcialmente, pois antes de implementar esta melhoria deve-se verificar o custo benefício da inclusão de mais uma atividade dentro do processo de desenvolvimento.

- Todos os participantes concordam plenamente que a melhoria sobre conscientizar os desenvolvedores sobre a importância do teste unitário deve ser aplicada, pois o teste unitário faz parte das atividades do desenvolvedor, e esta descrita no processo de desenvolvimento, porém é preciso reforçar e melhorar essa questão.

A escala Likert exibida na Tabela 1, foi utilizada para verificar o grau de concordância dos participantes, onde 1 é “discordo totalmente” e 5 “concordo totalmente”. Os valores menores que 3 são considerados como discordantes e os maiores que 3, como concordantes, considerando uma escala de 5 pontos. O valor igual a 3 é considerado “indiferente” ou “sem opinião”. Cada coluna da escala possui o número de participantes que responderam a questão para aquela escala. Para as questões utilizou-se a moda, que é o valor que mais se repete, ou seja, o valor mais provável a ser escolhido. Quando nenhum dos valores da escala se repetem, não possui moda, como a questão 3 da Tabela 1. O valor das questões que mais se repetiu foi o 5, indicando que os participantes tendem a concordar com as melhorias propostas para o processo de teste da Organização.

Tabela 1. Escala Likert

Questão	5. Concordo totalmente	4. Concordo parcialmente	3. Não sei opinar	2. Discordo parcialmente	1. Discordo totalmente	Moda
1	3	0	0	0	0	5
2	2	1	0	0	0	5
3	0	1	0	1	1	Nenhuma
4	3	0	0	0	0	5
5	2	0	0	1	0	5
6	3	0	0	0	0	5
7	3	0	0	0	0	5
8	3	0	0	0	0	5
9	2	1	0	0	0	5
10	2	0	0	0	1	5
11	3	0	0	0	0	5
12	2	0	0	0	1	5
13	3	0	0	0	0	5
14	3	0	0	0	0	5
15	3	0	0	0	0	5
16	3	0	0	0	0	5
17	3	0	0	0	0	5
18	2	1	0	0	0	5
19	3	0	0	0	0	5

5. Conclusão

Com a avaliação do processo de teste da Organização ABC utilizando o modelo *Test Process Improvement* – TPI, foi comprovado que a necessidade de melhoria do processo de teste é um fator importante para garantir a qualidade do software desenvolvido.

Com o resultado da avaliação foi possível identificar os pontos fortes e fracos atuais do processo de teste da Organização que precisam ser melhorados. A matriz de maturidade dos testes mostrou que o processo de teste da Organização não se encontra no nível Controlado, pois muitas áreas-chave não conseguiram alcançar o nível A de maturidade. O processo de teste precisa de algumas melhorias para elevar o nível de maturidade de acordo com o modelo TPI e ter seu processo de teste melhorado, e para isso foram apresentadas sugestões de melhorias por área-chave para a Organização.

As melhorias propostas foram elaboradas com base nas sugestões indicadas pelo modelo TPI e na própria Organização, sugerindo melhorias que fossem mais adequadas. Os Analistas de testes e Coordenador de equipe concordaram com grande parte das melhorias propostas para serem aplicadas na Organização.

Para solucionar problemas no processo de teste, o *Test Process Improvement* – TPI mostrou ser um modelo bastante completo, que abrange todas as áreas consideradas importantes para o teste de software e pode ser aplicado em qualquer Organização que precisa ter seu processo de teste melhorado.

Alguns pontos do modelo TPI não foram compatíveis com o processo de teste da Organização. Como exemplo, na área-chave Testes de baixo nível, para alcançar os níveis de maturidade, o modelo TPI sugere que as etapas de planejamento, especificação e execução sejam realizadas assim como nos testes de alto nível. Na Organização não seria aplicável elaborar casos de teste para testes unitários na fase de especificação, então foi sugerido conscientizar os desenvolvedores para realizar o teste unitário, considerando esta a melhoria mais adequada para a Organização.

Com base no estudo de caso realizado pode-se afirmar que com maior maturidade no processo de teste, é possível entregar um software mais confiável para o cliente. Neste caso, o que sempre deverá ser levado em consideração é que se não houver um processo maduro de teste, é provável que haja aumento dos prazos, custos e defeitos em produção.

Como contribuição, espera-se que o estudo de caso realizado possa auxiliar as organizações que precisam avaliar e melhorar seus processos de teste. O conhecimento sobre o modelo *Test Process Improvement* – TPI também é uma contribuição para empresas que desejam escolher um modelo de maturidade de testes. Para a área de testes de software este trabalho é de grande importância como fonte de aprendizagem sobre o TPI, a fim de auxiliar trabalhos de pesquisa ou enriquecer o conhecimento, pois este modelo não possui muitos documentos ou artigos como referência.

Para trabalhos futuros recomenda-se implementar as melhorias propostas para que a Organização tenha seu processo de teste melhorado e esteja sempre em melhoria contínua para ter seu processo de teste mais adequado e produtos com maior qualidade.

6. Referências

- Amorim, V. M. (2015) “Melhorando processos: Usando Ciclo de Deming – PDCA”
<http://www.devmedia.com.br/melhorando-processos-usando-ciclo-de-deming-pdca/22636>
- Araújo, A. F. (2013) “Um Arcabouço para Avaliação do Nível de Maturidade em Teste de Software para Micro e Pequenas Empresas”
http://www.portal.inf.ufg.br/mestrado/sites/www.inf.ufg.br.mestrado/files/uploads/DiSSERTACOES/adailton_ferreira_de_araujo.pdf
- Bastos, A.; Rios, E.; Cristalli, R. e Moreira, T. (2007) “Base de conhecimento em teste de software”, São Paulo, Martins.
- Cruz, G. A. (2010) “Avaliação de maturidade em Processo de Teste de software: Uma Pesquisa-ação”
http://repositorio.ufla.br/bitstream/1/5093/1/MONOGRAFIA_%20Avalia%C3%A7%C3%A3o%20de%20maturidade%20em%20processo%20de%20teste%20de%20software%3A%20uma%20pesquisa-a%C3%A7%C3%A3o.pdf
- Gil, A. C. (2010) “Como Elaborar Projetos de Pesquisa”, São Paulo, Atlas.
- Gressler, L. A. (2004) “Introdução à pesquisa: projetos e relatórios / Lori Alice Gressler.”, São Paulo, Loyola, 295 p.
- Koomen, T. e Pol, M. et al.(1999) “Test Process Improvement: A Practical Step-by-Step Guide to structured testing.”, England, Addison-Wesley.
- Malhotra, N. K. (2012) "Pesquisa de Marketing uma Orientação Aplicada", Bookman, 6ª edição
- Pressman, R. S. (2011), “Engenharia de software”, Bookman, 7ª edição
- Richardson, R. J. (1999) “Pesquisa social: métodos e técnicas”, São Paulo, Atlas, 3ª edição
- Rios, E. e Moreira, T. (2013) “Teste de Software”, Rio de Janeiro, Alta Books, 3ª edição.
- Softexrecife (2011) “MPT Melhoria do Processo de Teste Brasileiro. Guia de Referência do Modelo – MPT.Br.” http://mpt.org.br/mpt/wp-content/uploads/2013/05/MPT_Guia_de_referencia.pdf.
- Sommerville, I. (2005) “Engenharia de Software”, Addison Weasley, 6ª edição.
- Veenendaal, E. (2012) “TMMi Foundation. Test Maturity Model Integration (TMMi)”
<http://www.tmmi.org/pdf/TMMi.Framework.pdf>.
- Wainer, J. (2007) “Métodos de Pesquisa Quantitativa e Qualitativa para a Ciência da Computação” <http://www.ic.unicamp.br/~wainer/papers/metod07.pdf>
- Yin, R. K. (2001) “Estudo de Caso. Planejamento e Métodos”, Porto Alegre, Bookman, 2ª edição.

ANEXO A

1 Estratégia de teste					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
1. A Estratégia de teste individuais para Teste de alto nível					5A, 11A
1. A.1	Os riscos do produto são considerados. O conhecimento do sistema, seu uso e gerenciamento operacional são necessários.	x			
1. A.2	Há uma diferenciação na profundidade dos testes dependendo dos riscos, e se existirem, nos critérios de aceitação: nem todos os subsistemas e características de qualidade são testadas igualmente nos sistemas.	x			
1. A.3	Uma ou mais técnicas de especificação de testes são usadas, adaptadas à profundidade exigida para o teste.	x			
1. A.4	Para retestes existe uma estratégia simples para decidir entre testar somente os problemas resolvidos ou retestar tudo.		x		
1. B Estratégias combinadas para testes de alto nível					2A, 5B, 11B, 14B, 18B
1. B.1	A coordenação ocorre entre os diferentes testes de alto nível, entre eles: sistema, aceitação e testes de aceitação em produção, no campo da estratégia de teste (riscos, características de qualidade, área de consideração do teste, e planejamento).		x		
1. B.2	O resultado da coordenação é uma estratégia coordenada, e documentada. Durante todo o processo teste, esta estratégia é controlada.		x		
1. B.3	Cada teste de alto nível determina a sua própria estratégia de teste, com base na estratégia de coordenação, tal como é descrito no nível A.		x		
1. B.4	Os desvios em relação à estratégia de coordenação são relatados, após o qual é feito um ajuste fundamentado à estratégia de coordenação, com base nos riscos.		x		
1. C Estratégias combinadas para testes de alto nível e Testes de baixo nível ou avaliação					3C, 19B ou 20C
1. C.1	A coordenação ocorre nos testes de alto nível e em testes de baixo nível ou em níveis de avaliação na área de estratégia de teste (riscos, características de qualidade, área de consideração do teste / avaliação e planejamento).		x		
1. C.2	O resultado da coordenação é uma estratégia coordenada, na qual é documentada. Durante todo o processo de teste (e avaliação) essa estratégia é controlada.		x		
1. C.3	Cada teste de alto nível determina, com base na coordenação, a sua estratégia de teste, como descrito na seção 1. A		x		
1. C.4	Cada teste de baixo nível determina, com base na coordenação, a sua estratégia de teste, como descrito na área chave "Testes de baixo nível" (nível C).		x		
1. C.5	Cada nível de avaliação determina, com base na coordenação, a sua estratégia de avaliação, como é descrito na área chave "Avaliação"(nível B).		x		
1. C.6	Os desvios em relação à estratégia de coordenação são relatados, e com base nos riscos são realizados ajustes fundamentados na estratégia de coordenação.		x		

1. D Estratégias combinadas para todos os níveis de teste e avaliação				3C, 19B, 20C	
1. D.1	A coordenação ocorre nos testes de alto nível, testes de baixo nível e em níveis de avaliação na área de estratégia de teste (riscos, características de qualidade, área de consideração do teste / avaliação e planejamento).		x		
1. D.2	O resultado da coordenação é uma estratégia de coordenação, e está documentada. Durante todo o processo de avaliação e teste, esta estratégia é controlada.		x		
1. D.3	Cada teste de alto nível determina a sua própria estratégia baseando-se na coordenação, como descrito no nível A		x		
1. D.4	Cada teste de baixo nível determina a sua própria estratégia de teste baseando-se na coordenação, como descrito na área-chave "Teste de baixo nível" (nível C).		x		
1. D.5	Cada nível de avaliação determina a sua própria estratégia de avaliação baseando-se na coordenação, como descrito na área-chave "Avaliação" (nível B).		x		
1. D.6	Os desvios em relação à estratégia de coordenação são relatados, e com base nos riscos são realizados ajustes fundamentados na estratégia de coordenação.		x		
2 Modelo de ciclo de vida					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
2. A Planejamento, Especificação e Execução					11A
2. A.1	Para o teste, no mínimo, as seguintes fases devem ser reconhecidas formalmente: Planejamento, Especificação e Execução de testes. Estas fases devem ser realizadas, possivelmente por subsistema. Uma certa sobreposição entre elas é permitida.	x			
2. A.2	Atividades a serem executadas na fase de Planejamento:				
2. A.2.1	formulação das tarefas;	x			
2. A.2.2	determinar todos os documentos que servirão de base para o teste;	x			
2. A.2.3	definir a estratégia de teste;	x			
2. A.2.4	fazer um plano detalhado dos testes, ou seja, estabelecer a organização do teste;	x			
2. A.2.5	estabelecer quais serão os entregáveis do teste;	x			
2. A.2.6	definir a infra-estrutura e as ferramentas;	x			
2. A.2.7	estabelecer como será o gerenciamento do teste;	x			
2. A.2.8	definir o planejamento;	x			
2. A.2.9	elaborar o plano de teste;		x		
2. A.3	Atividades a serem executadas na fase de Especificação:				
2. A.3.1	elaborar e revisar os casos de testes, scripts/roteiros;	x			
2. A.3.2	verificar objetos de teste e infra-estrutura	x			
2. A.3.3	preparar a infra-estrutura do teste (ferramentas e ambiente de teste);	x			
2. A.4	Atividades a serem executadas na fase de Execução:				
2. A.4.1	realizar teste para avaliar a infra-estrutura;	x			
2. A.4.2	carregar as bases de dados iniciais dos teste;	x			
2. A.4.3	executar o teste e reteste.	x			

2. B Planejamento, Preparação, Especificação, Execução e Conclusão.					6A, 17A
2. B.1	Para o teste, no mínimo, as seguintes fases devem ser reconhecidas formalmente: Planejamento, Preparação, Especificação, Execução e Conclusão de testes. Estas fases devem ser realizadas, possivelmente por subsistema. Uma certa sobreposição entre elas é permitida.	x			
2.B.2	Atividades a serem executados em cada fase:				
2. B.2.1	Preparação: inspecionar a documentação na qual os teste se baseiam (verificar se a documentação é apropriada para as técnicas de testes escolhidas).	x			
2. B.2.2	Conclusão: Atualizar o testware (de forma que possa ser reutilizado em outros processos de teste); avaliar o objeto de testes, avaliar o processo de teste; e fazer o reporte final.	x			
3 Momento do envolvimento					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
3. A Conclusão da documentação que serve de base para o teste					2A
3. A.1	A atividade de testes começa com a conclusão ou até antes da conclusão da documentação que serve de base para o teste de uma parte restrita do sistema que será testado separadamente. O sistema pode ser dividido em várias partes que podem ser construídas, finalizadas e testadas separadamente. A atividade de teste do primeiro subsistema (ou módulo) tem que começar ao mesmo tempo ou antes da finalização da documentação que serve de base para o teste.	x			x
3. B Início da documentação que serve de base para o teste					2B
3. B.1	A atividade de testes começa com a definição da documentação ou até antes da definição da documentação que serve de base para o teste (especificações funcionais).		x		
3.C Início da definição de requisitos					
3. C.1	A atividade de testes começa com a definição de requisitos ou até antes da definição de requisitos.		x		
3. D Início do projeto					11C
3. D.1	Quando o projeto é iniciado, começa também a atividade de testes.		x		
4 Planejamento e Estimativa					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
4. A O planejamento e estimativa são substanciados					2A
4. A.1	A estimativa e planejamento do teste podem ser substanciados (e não somente "nós fizemos isto deste modo no último projeto").	x			x
4. A.2	No processo de teste, o planejamento e estimativa são monitorados, os ajustes são feitos se necessário.	x			x
4. B O planejamento e estimativa são substanciados estatisticamente					7B, 15B
4. B.1	As métricas relativas ao progresso e qualidade são estruturalmente mantidas para vários projetos (Nível B da área chave Métricas).		x		
4. B.2	Os dados das métricas são utilizados para substanciar a estimativa e planejamento de teste.		x		

5 Técnicas de especificação de testes					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
5. A Técnicas Informais					
5. A.1	Os casos de teste são definidos de acordo com uma técnica documentada.	x			
5. A.2	A técnica consiste no mínimo em: a) situação inicial, b) processo de mudança = ações de teste a serem executadas, c) resultado esperado.	x			
5. B Técnicas formais					12A, 17A
5. B.1	Além das técnicas informais, técnicas formais também são utilizadas, provendo de forma não ambígua os casos de testes elaborados a partir da documentação que serve de base para os testes		x		
5. B.2	Um julgamento substanciado sobre o nível de cobertura da coleção dos casos de teste é possível (comparando com a documentação que serve de base para os testes básicos).	x			
5. B.3	Os produtos gerados no processo de testes "testware" (scripts, especificações de testes, descrição da infra-estrutura, etc) são reutilizáveis dentro do time de testes, ou seja, a uniformização do método de trabalho permite a reutilização dos produtos.	x			
6 Técnicas estáticas de teste					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
6. A Inspeção da documentação utilizada como base para o teste					
6. A.1	Antes de definir os casos de testes, são realizados estudos da testabilidade da documentação utilizada como base para os testes.		x		
6. A.2	Os estudos são feitos através de <i>Checklists</i> .		x		
6. B Checklists					
6. B.1	Outros testes estáticos além da inspeção da documentação utilizada como base para teste são realizados através de <i>checklists</i> (aprovado pelo projeto e/ou cliente).		x		
7 Métricas					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
7. A Métricas do projeto de testes (produtos)					11B, 15B, 16A, 18B
7. A.1	As seguintes métricas de entrada no projeto de testes são registradas:				
7. A.1.1	Quantidade de recursos e horas utilizadas por recurso.	x			
7. A.1.2	Atividades executadas: horas previstas X horas realizadas por atividade.	x			
7. A.1.3	Tamanho e complexidade do sistema testado: em pontos de função, número de funções e/ou esforço de construção.	x			
7. A.2	As seguintes métricas de saída do projeto de testes são registradas:				
7. A.2.1	Produtos de teste: quantidade de especificações e casos de teste, quantidade de relatórios de log.		x		
7. A.2.2	Progresso do teste: quantos e quais testes executados, status de cada teste (concluído / não concluído)		x		
7. A.2.3	Número de defeitos: defeitos por nível de teste (unitário, sistema, aceitação, etc), por subsistema, por motivo, prioridade, status (novo, em solução, corrigido, re-testado)		x		
7. A.3	As métricas são utilizadas no reporte dos testes.		x		

7. B Métricas do projeto de testes (processo)				15C, 16B	
7. B.1	No projeto de teste as medições são feitas por pelo menos dois itens mencionados abaixo:		x		
	- Eficácia de detecção de defeitos: defeitos encontrados no teste X defeitos encontrados em produção, qual teste deveria ter descoberto o defeito.		x		
	- Eficiência na detecção de defeitos: defeitos encontrados por hora no teste.		x		
	- Nível de cobertura de teste: com base em linhas de código, especificações funcionais, condições possíveis.		x		
	- Qualidade Testware: número de defeitos encontrados por "testar errado", em comparação com o número total de defeitos encontrados.		x		
	- Percepção da qualidade (com base em entrevistas com usuários, testadores e pessoas envolvidas no desenvolvimento)		x		
7. B.2	As métricas são utilizadas no reporte dos testes.		x		
7. C Métricas de sistema				13B, 14C, 18C	
7. C.1	As métricas mencionadas acima são registradas para o desenvolvimento, manutenção e de produção.		x		
7. C.2	As métricas são utilizadas na avaliação da eficácia e eficiência do processo de teste.		x		
7. D Métricas da organização					
7. D.1	As métricas mutuamente comparáveis são mantidas para os dados já mencionados.		x		
7. D.2	A métricas são usadas para avaliar a eficácia e eficiência dos processos de teste separados para obter uma otimização da metodologia de teste genérica e processos de testes futuros.		x		
8 Ferramentas de teste					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
8. A Ferramentas de controle de planejamento					
8. A.1	Ferramentas automatizadas (com exceção de ferramentas de texto) são utilizadas para a administração de defeitos e pelo menos outras duas para atividades de planejamento e controle.	x			
8. B Ferramentas de análise e execução				5B, 12A	
8. B.1	Pelo menos dois tipos de ferramentas automatizadas são utilizadas para a execução de teste, tais como ferramentas de captura e reprodução, ferramentas de cobertura de teste, etc.			x	
8. B.2	A equipe de testes tem uma visão geral sobre a relação custo/benefício dessas ferramentas.		x		
8. C Automação extensa no processo de teste					
8. C.1	Ferramentas automatizadas são utilizadas na a fase de planejamento (estimativa, planejamento, monitoramento do progresso, gerenciamento da configuração e administração de defeitos), preparação, especificação e execução. No total, pelo menos, cinco tipos de ferramentas devem estar em uso.			x	
8. C.2	A equipe de teste tem uma visão sobre a relação custo/benefício dessas ferramentas.		x		

9 Ambiente de testes					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
9. A Ambiente de teste gerenciado e controlado					12A
9. A.1	Somente com autorização do Gerente de testes pode haver alterações no ambiente ou troca de versões no ambiente de testes		x		
9. A.2	O ambiente deve ser disponibilizado em tempo hábil.	x			
9. A.3	O ambiente de teste é gerenciado com respeito a configuração disponibilidade, manutenção, gerenciamento de versão, tratamento de erros, autorizações, etc.	x			
9. A.4	Salvas e recuperações de certas situações de testes podem ser feitas rapidamente e facilmente.	x			
9. A.5	O ambiente é suficientemente representativo para permitir a execução dos testes. Quanto mais próximo da produção for o teste, mais representativo dever ser o ambiente de teste em relação ao de produção.	x			
9. B Testes no ambiente mais adequado					1B
9. B.1	Cada teste é realizado no ambiente mais adequado, pela execução em outro ambiente ou por adaptar rapidamente e facilmente o próprio ambiente.	x			
9. B.2	O ambiente é finalizado em tempo hábil para o teste e não há perturbação por outras atividades durante o teste.	x			
9. B.3	Os riscos assumidos são analisados e são tomadas as medidas adequadas.	x			
9. C Ambiente adequado e a tempo					
9. C.1	O ambiente é o mais adequado para o teste, é muito flexível e pode ser rapidamente adaptado para mudança de requisitos.	x			
10 Local de trabalho					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
10. A Local de trabalho adequado e disponibilizado a tempo					
10. A.1	A infra-estrutura necessária do local de trabalho dos testadores (escritórios, salas, telefones, PCs, conexões de rede, software, impressoras, e etc) é disponibilizada a tempo.	x			
10. A.2	Aspectos relacionados com a organização da infra-estrutura do local de trabalho tem um impacto mínimo no processo de teste (a distância entre os testadores e o resto do projeto não é muito grande).	x			
11 Motivação e comprometimento					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
11. A Definição do orçamento e tempo					
11. A.1	O teste é considerado pelo pessoal envolvido como necessário e importante.	x			
11. A.2	É disponibilizado tempo e verba para atividades de teste.	x			
11. A.3	O controle do gerenciamento do teste é baseado em tempo e dinheiro. Uma característica é que se o tempo ou orçamento são excedidos, inicialmente procura-se uma solução (fazendo horas extra ou contratando mais pessoas quando os limites de tempo forem excedidos ou diminuindo o tempo e/ou orçamento)	x			
11. A.4	Na equipe há bastante conhecimento e experiência na área de testes.	x			

11. A.5	A maioria dos participantes participa das atividades de teste em tempo integral, portando não há muitos conflitos com outras atividades.	x			
11. A.6	Há uma boa relação entre os testadores e outras áreas do projeto e da organização.	x			
11. B Teste integrado com a organização do projeto					2A, 15B, 16A, 18B
11. B.1	Todos os envolvidos acham que o teste tem uma influência positiva e notável na qualidade do produto.	x			x
11. B.2	O gerenciamento do teste procura ter uma visão sobre a profundidade e qualidade dos testes.	x			x
11. B.3	O gerenciamento do controle dos testes é baseado em tempo, dinheiro e qualidade. Uma característica é que as soluções para os problemas no teste (por exemplo: exceder o tempo ou orçamento) também são procuradas fora do projeto de teste. Possivelmente o desenvolvedor poderá ser acionado aqui.	x			x
11. B.4	No planejamento do projeto são considerados o ciclo de teste, retrabalho e reteste.	x			x
11. B.5	A equipe de teste se comunica com o desenvolvedor na entrega de versões.	x			x
11. B.6	Conselhos da área de testes são discutidos em reuniões do projeto.	x			x
11. C Engenharia de teste					1C, 3C, 8B, 15C
11. C.1	A equipe de teste está envolvida na concepção e realização de fornecer a capacidade ideal de teste do sistema ("design para teste").	x			
11. C.2	A equipe de teste tem conhecimento e habilidades suficientes para fornecer uma realização significativa do ponto de verificação mencionado acima (11. C.1).	x			
11. C.3	As recomendações da equipe de teste são consideradas seriamente pela Organização e / ou projeto.	x			
11. C.4	A gestão apoia os testadores (com pessoas e meios) em trabalhar continuamente na melhoria do processo de teste.	x			
11. C.5	Participação em teste é considerada como uma "promoção"; a atividade de teste tem um status alto.		x		
11. C.6	O processo de desenvolvimento tem maturidade suficiente; pelo menos, tempo e qualidade são controlados.	x			
11. C.7	A carreira em teste está descrita no nível da organização, incluindo possibilidade de carreira e gratificação.	x			
12 Atividades e treinamento de testes					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
12. A Gerente de testes e testadores					
12. A.1	A equipe de testes é composta de no mínimo um gerente de teste e testadores.			x	
12. A.2	As tarefas e responsabilidades são definidas.	x			
12. A.3	A equipe de teste teve treinamento específico em testes (por exemplo: gerenciamento de testes, técnicas de teste, etc) ou tem experiência suficiente na área de testes.			x	
12. A.4	Para o teste de aceitação, o conhecimento necessário para realizar o		x		

	teste é disponibilizado para a equipe de testes.				
12. B Suporte metódico, técnico e funcional. Gestão do processo de teste, testware e infra-estrutura.					
12. B.1	Há suporte disponível para ajudar os testadores a compreender e aplicar a metodologia de teste.		x		
12. B.2	Há suporte técnico disponível para manter o ambiente de teste e as ferramentas necessárias para os testadores.	x			
12. B.3	O suporte das pessoas que definiram e implementaram as funcionalidades que estão sendo testadas é disponível.	x			
12. B.4	O gerenciamento do processo de teste é descrito separadamente e é responsável pelo registro, armazenamento e disponibilização de todos os objetos do processo de teste de gestão. Objetos a serem administrados são de progresso, orçamentos e defeitos.		x		
12. B.5	Há um processo separado, para o gerenciamento do testware (documentação, documento que serve de base para o teste, casos de teste, arquivos de entrada de teste e bancos de dados, instruções de teste e procedimentos).	x			
12. B.6	Há um processo separado, para o gerenciamento do ambiente de teste.	x			
12. B.7	A pessoa que realiza essas tarefas têm conhecimento e experiência suficientes.	x			
12. B.8	O tempo necessário para estas tarefas está previsto. É realizada uma supervisão para verificar que essas tarefas estão de fato sendo executadas.		x		
12.C Garantia da qualidade interna formal					13A
12. C.1	Paralelamente ao plano de teste, é formulado um plano (QA) de garantia da qualidade interno.		x		
12. C.2	A pessoa atribuída a tarefa de QA (garantia da qualidade) não tem outras tarefas dentro da equipe de teste.		x		
12. C.3	Os resultados das atividades de garantia da qualidade (QA) são utilizados para a melhoria do processo de teste.		x		
12. C.4	A pessoa que executa a tarefa de garantia da qualidade (QA) tem conhecimento suficiente e experiência.		x		
13 Escopo da metodologia					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
13. A Específica por Projeto					2A, 5B, 16A, 17A, 18B
13. A.1	Uma metodologia é formulada para cada projeto.		x		
13. A.2	A metodologia cobre no mínimo os seguintes itens: uma descrição do ciclo de vida de teste, a gestão do processo de teste, a gerenciamento do produtos de teste, e a especificação de técnicas de teste utilizadas.		x		
13. A.3	A metodologia é seguida.		x		
13. B Organização genérica					
13. B.1	A metodologia é definida em um modelo genérico para a organização.		x		
13. B.2	Cada projeto funciona de acordo com este modelo genérico.		x		
13. B.3	Os desvios são suficientemente discutidos e documentados.		x		

13. C Otimização da organização e atividades com Rapid Application Development - RAD(Desenvolvimento Rápido de Aplicação)					11B, 18C
13. C.1	Há um processo de feedback estruturado (formal e acionado pelo departamento RAD-Desenvolvimento Rápido de Aplicação) no modelo genérico.		x		
13. C.2	A manutenção estrutural e de inovação são feitas sobre o modelo genérico, por exemplo, sobre a base de feedback.		x		
14 Comunicação					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
14. A Comunicação interna					
14. A.1	Há uma reunião periódica com a equipe de teste. Esta reunião possui uma agenda fixa e o principal foco é o progresso e a qualidade do objeto a ser testado.			x	
14. A.2	Periodicamente, cada membro da equipe participa da reunião.			x	
14. A.3	Os desvios do plano de teste são comunicados e documentados.		x		
14. B Comunicações no Projeto					2A, 15B, 16A
14. B.1	Na reunião da equipe de teste é produzida uma ata.		x		
14. B.2	Na reunião da equipe de teste, além de progresso e a qualidade do objeto de teste, a qualidade do processo de teste também é pauta fixa das reuniões.		x		
14. B.3	Periodicamente, o gerente de teste reporta na reunião do projeto o progresso e a qualidade do objeto a ser testado, incluindo os riscos. O gerente de testes também reporta a qualidade do processo de teste.		x		
14. B.4	Os acordos desta reunião são documentados.		x		
14. B.5	O gerente de teste é informado em tempo sobre as mudanças nas datas de entrega previstas e anteriormente acordadas (tanto na documentação que serve de base para o teste quanto nos objetos).	x			
14. B.6	Na reunião periódica sobre os defeitos, as soluções para os defeitos são discutidos entre os representantes da equipe de teste e de outras partes envolvidas.		x		
14. B.7	A equipe de testes está envolvida no controle de alterações para julgar o impacto que as alterações irão provocar no esforço de teste.	x			
14. C Comunicação dentro da Organização sobre a qualidade dos processos de teste					13B
14. C.1	Há uma reunião periódica onde propostas para melhoria da metodologia de teste utilizada são discutidas.		x		
14. C.2	Os participantes são representantes das equipes e área de teste.		x		
15 Relatórios					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
15. A. Defeitos					
15. A.1	Os defeitos encontrados são reportados periodicamente, e classificados como defeitos resolvidos e não resolvidos.		x		
15. B Progresso, Atividades, Defeitos com as prioridades					2A, 16A, 18B
15. B.1	Os defeitos são reportados e classificados por severidade de acordo com normas claras e objetivas.		x		

16. C Gerenciamento de defeitos do projeto					
16. C.1	A administração de defeitos é utilizada integralmente no projeto. Os defeitos são registrados pelos envolvidos no desenvolvimento do sistema e podem ser administrados por eles mesmos.	x			
16. C.2	As autorizações garantem que cada usuário da administração poderá fazer somente o que tem permissão.	x			
17 Gerenciamento dos produtos de teste					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
17. A Gerenciamento dos produtos de testes internos					
17. A.1	Os produtos de testes (casos de teste, inicialização de base de dados, etc) documentação que serve como base para o teste, objetos de teste, documentação de teste e as diretrizes de teste são gerenciados internamente de acordo com um procedimento descrito, contendo medidas para a entrega, registro, arquivamento.	x			
17. A.2	O gerenciamento de versões considera os relacionamentos entre várias partes: documentação que serve de base para o teste, objeto de teste, testware, etc.	x			
17. A.3	A transferência de versões e documentos para o time de testes ocorre de acordo com um procedimento padrão. As partes que constituem uma transferência devem ser: quais partes e versões do objeto de teste, que versão da documentação que serve de base para o teste, defeitos resolvidos, defeitos não corrigidos, incluindo os defeitos do próprio desenvolvedor.	x			
17. B Gerenciamento externo da documentação que serve de base para o teste e do objeto de teste					
17. B.1	A documentação que serve de base para teste e o objeto de teste (geralmente desenho e software) tem um gerenciamento de versões de acordo com um procedimento descrito, com passos para a entrega, registro e arquivamento.	x			
17. B.2	O gerenciamento de versões considera os relacionamentos entre várias partes (documentação que serve de base para teste e o objeto de teste).	x			
17. B.3	A equipe de teste é informada sobre mudanças na documentação que serve de base para teste ou no objeto de teste em tempo hábil.	x			
17. C Os produtos de teste são reutilizáveis					
					5B
7. C.1	Os produtos de teste (ou um subconjunto acordados) são finalizados e após a conclusão dos testes são transferidos para a área de manutenção formalmente acordados.	x			x
7. C.2	Os produtos de teste transferidos são reutilizados.	x			x
17. D Rastreabilidade dos requisitos do sistema para os casos de teste					
7. D.1	Cada requisito e especificação do sistema estão relacionados a um ou mais casos de teste de forma transparente, e vice-versa.	x			
7. D.2	Estas relações são rastreáveis através de versões separadas (por exemplo, o requisito do sistema A, versão 1.0, está relacionado com design funcional B, versão 1.3 e com programas C e D, as versões 2.5 e 2.7, e estão relacionados com os casos de teste X a Z, versão 1.4.	x			

18 Gerenciamento do processo de teste					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
18. A Planejamento e execução					
18. A.1	Antes das atividades de teste, um plano de teste é formulado e são mencionados todas as atividades a serem executadas. Para cada atividade há uma indicação do período no qual ocorre, os recursos (pessoas ou meios necessários) e os produtos a serem entregues.			x	
18. B Planejamento, Execução, Monitoramento e Ajuste					
18. B.1	Há monitoramento de execução de todas as atividades planejadas.	x			
18. B.2	Cada atividade também é monitorada em termos de tempo e dinheiro.	x			
18. B.3	Os desvios são documentados.		x		
18. B.4	No caso de desvios são feitos ajustes, ajustando o plano, ou realizando atividades novamente de acordo com o plano. O ajuste é substanciado.		x		
18. C Monitoramento e ajuste do processo de testes na Organização					13B
18. C.1	Existe um acompanhamento da aplicação da metodologia (métodos, normas técnicas e procedimentos) da organização.		x		
18. C.2	Os desvios são documentados e comunicados ao processo de teste.		x		
18. C.3	Em caso de desvios os riscos são analisados e os ajustes são feitos, por exemplo, adaptando a metodologia, as atividades ou os produtos para se adequarem a metodologia. O ajuste é substanciado.		x		
19 Avaliação					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
19. A Técnicas de avaliação					
19. A.1	Na avaliação dos produtos (especificação de requisitos) são utilizadas técnicas.		x		
19. A.2	A avaliação e os seus resultados são reportados.		x		
19. A.3	A manipulação dos resultados é monitorada.		x		
19. A.4	Os testadores estão envolvidos nestas avaliações.		x		
19. B Estratégia de avaliação					
19. B.1	A análise de riscos é considerada na avaliação.		x		
19. B.2	Existe uma diferenciação na análise e profundidade das avaliações de acordo com os riscos e critérios de aceitação. Nem todos os produtos intermediários são igualmente avaliados.		x		
19. B.3	As escolhas são feitas a partir de várias técnicas de avaliação, apropriadas a profundidade da avaliação.		x		
19. B.4	Para re-avaliação existe uma simples estratégia que consistem em escolher entre "avaliar apenas problemas resolvidos" e "fazer uma re-avaliação completa".		x		
19. B.5	A estratégia é determinada e executada posteriormente. É verificado se a execução das avaliações é realizada de acordo com a estratégia e se necessário são feitos ajustes.		x		

20 Teste de baixo nível					
Pontos de verificação		Sim	Não	Parcialmente	Dependências
20. A Modelo de ciclo de vida dos testes de baixo nível (Planejamento, especificação e execução)					
20. A.1	Para testes de baixo nível, no mínimo, estas fases são identificadas: planejamento, especificação e execução. São executadas nesta sequência e para cada subsistema ou módulo, se aplicável.			x	
20. A.2	Atividades a serem executados em cada fase:				
20. A.2.1	Planejamento: formulação das tarefas; determinar todos os documentos que servirão de base para o teste; definir a estratégia de teste; fazer um plano detalhado dos testes, ou seja, estabelecer a organização do teste; configurar os produtos de teste; definir a infraestrutura e as ferramentas; estabelecer como será o gerenciamento do teste; definir o planejamento; elaborar o plano de teste;		x		
20. A.2.2	Especificação: elaborar os casos de testes, scripts/roteiros; preparar a infraestrutura do teste (ferramentas e ambiente de teste)		x		
20. A.2.3	Execução: carregar as bases de dados iniciais dos testes; executar o teste e reteste.	x			
20. B Técnicas de caixa branca					
20. B.1	Além de técnicas informais, os testes de baixo nível também usam técnicas formais, proporcionando uma rota clara desde a documentação que serve como base para o teste até os casos de teste.		x		
20. B.2	Para os testes de baixo nível, é possível fazer uma declaração sobre o nível de cobertura de testes, em comparação com a documentação que serve como base para o teste.		x		
20. B.3	O testware é reutilizável (com a equipe de testes) por um método de trabalho uniforme.		x		
20. C Estratégias de teste de baixo nível					
20. C.1	Os riscos do produto são considerados. O conhecimento do sistema, seu uso e gerenciamento operacional são necessários.		x		
20. C.2	Há uma diferenciação na profundidade dos testes dependendo dos riscos, e se existirem, nos critérios de aceitação: nem todos os subsistemas e características de qualidade são testadas igualmente nos sistemas.		x		
20. C.3	Uma ou mais técnicas de especificação de testes são usadas, adaptadas à profundidade exigida para o teste.		x		
20. C.4	Para retestes existe uma estratégia simples para decidir entre testar somente os problemas resolvidos ou retestar tudo.		x		
20. C.5	A estratégia é determinada e posteriormente executada. Verifica-se se a execução dos testes estão de acordo com a estratégia. Se necessário são feitos os ajustes.		x		

APÊNDICE A

Validação das Melhorias do processo de testes
As melhorias propostas foram indicadas através do resultado da avaliação do processo de testes que foi realizada com Analistas de Teste da Organização. Para avaliação do processo de teste foi utilizado o modelo <i>Test Process Improvement</i> – TPI.
Preencher o formulário marcando com um X nas seguintes opções: Concordo totalmente: Concorda com a melhoria proposta, isto é, é adequada para a Organização e pode ser implementada. Informar a justificativa se necessário. Concordo parcialmente: Concorda com a melhoria proposta com algumas restrições. Informar a justificativa. Discordo parcialmente: Não concorda com alguns pontos sobre a melhoria. Informar a justificativa. Discordo totalmente: Não concorda com a melhoria, pois não é adequada para a Organização. Informar a justificativa. Não sei opinar: Não tem conhecimento da área ou a proposta de melhoria.
Área chave: Estratégia de testes
1. Realizar testes de regressão para garantir que nenhuma parte do sistema deixou de funcionar após implementação ou manutenção do sistema.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
2. Ter uma estratégia de coordenação entre os diferentes níveis de teste como testes de sistema e unitário para definir qual a cobertura de testes e também os riscos envolvidos.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
3. Os testadores podem auxiliar no teste de aceitação, indicando para o cliente quais testes são mais importantes de serem executados com a ajuda dos casos de teste.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
Áreas chave: Modelo de ciclo de vida/ Motivação e comprometimento/ Gerenciamento do processo de teste
4. Elaborar um Plano de teste para projetos maiores e novas implementações do sistema.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
Área chave: Momento do envolvimento
5. Envolver o Analista de testes com experiência logo no início do projeto ou especificação de requisitos junto com Analista de sistemas e Analista de negócios para garantir que os requisitos mencionados estão corretos e/ou indicar novos.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
Áreas chave: Técnicas de especificação de testes / Gerenciamento dos produtos de teste
6. Utilizar uma técnica de especificação de casos de teste formal para elaborar casos de teste padronizados e com instruções detalhadas, de modo que o testador que irá executar o teste tenha informações suficientes para conduzir de forma correta e sem dúvidas o passo a passo do caso de teste escrito. A técnica deverá ser documentada e utilizada por todos os testadores da empresa.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
Área chave: Técnicas estáticas de teste
7. Utilizar uma técnica estática de teste para verificar possíveis erros na documentação do sistema. Uma das técnicas indicadas pelo TPI é baseada em <i>Checklists</i> , onde é formada uma equipe de inspeção, podendo ser os próprios testadores, utilizando uma lista com os itens a serem verificados e o que deve

ser entendido como defeito.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
Área chave: Métricas/ Planejamento e estimativa
8. Utilizar métricas de produto para avaliar a quantidade de defeitos que ocorreram entre as diferentes versões do sistema, quais as possíveis causas e ajudar na tomada de decisões para melhorar a qualidade do sistema. As métricas devem ser exibidas num relatório.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
Área chave: Ferramentas de teste
9. Visando maior agilidade nas atividades do processo de testes, utilizar uma nova ferramenta de testes poderia contribuir para consideráveis ganhos de tempo, produtividade, confiabilidade, qualidade na construção e execução dos casos de teste e em todo ciclo de vida do teste.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
Área chave: Motivação e comprometimento
10. Levar para o setor de Recursos Humanos a importância da área de testes dentro da Organização informando as vantagens e o papel importante que tem para que o produto seja entregue com qualidade, para que a área de teste tenha um status alto. Informar que através da avaliação do processo de teste com os profissionais da área, esta questão não foi alcançada positivamente.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
Área chave: Atividades e treinamento de teste
11. Realizar treinamentos para os testadores para qualificação na área de testes de software, negócio e o sistema da Organização.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
12. Para projetos mais críticos e com tempo de duração maiores, realizar testes de aceitação seria de grande importância para validar com os usuários se os requisitos estão de acordo com o que foi solicitado e também para a aprovação da solução entregue. Com o teste de aceitação o retrabalho, perdas, prejuízos e defeitos em produção seriam evitados. Disponibilizar para o cliente o conhecimento necessário para realizar os testes de aceitação com auxílio dos Analistas de testes para conduzir os testes no sistema.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
13. Ter um Analista de qualidade para auditar os procedimentos, atividades e o processo de testes e garantir que os processos estão sendo seguidos de acordo com os padrões e metodologia.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
Áreas chave: Escopo da metodologia/ Gerenciamento do processo de teste:
14. Definir e seguir uma metodologia de teste com o propósito de contribuir para o alcance da qualidade do sistema e ter um processo focado na melhoria do ciclo de desenvolvimento.
() 5. Concordo totalmente () 3. Não sei opinar () 1. Discordo totalmente () 4. Concordo parcialmente () 2. Discordo parcialmente
Justificativa:
Área chave: Comunicação
15. Realizar reuniões da equipe de teste com o coordenador da equipe para relatar o andamento das atividades, os problemas enfrentados no projeto, os impedimentos, o que precisa ser melhorado no processo de teste e reportar os defeitos mais críticos. Quando o projeto tem uma boa comunicação entre

os envolvidos, um plano de ação pode ser iniciado para solucionar os problemas, melhorias poderão ser implementadas e análise de risco. Documentar os pontos fracos e fortes da reunião e as ações de melhoria.
<input type="checkbox"/> 5. Concordo totalmente <input type="checkbox"/> 3. Não sei opinar <input type="checkbox"/> 1. Discordo totalmente <input type="checkbox"/> 4. Concordo parcialmente <input type="checkbox"/> 2. Discordo parcialmente
Justificativa:
Áreas chave: Relatórios/ Planejamento e estimativa/ Motivação e comprometimento
16. Enviar relatórios para os envolvidos no projeto e coordenador da equipe sobre a atual situação do projeto, defeitos e impedimentos, para dar uma visibilidade do andamento e progresso do projeto, os problemas que estão sendo enfrentados para a finalização e questões sobre o tempo disponível para a entrega do projeto.
<input type="checkbox"/> 5. Concordo totalmente <input type="checkbox"/> 3. Não sei opinar <input type="checkbox"/> 1. Discordo totalmente <input type="checkbox"/> 4. Concordo parcialmente <input type="checkbox"/> 2. Discordo parcialmente
Justificativa:
Área chave: Gerenciamento dos defeitos
17. Ter um acompanhamento do projeto e qualidade do sistema pelo coordenador da equipe, através da ferramenta Mantis utilizada pela equipe de testes, para verificar a quantidade de defeitos em aberto, resolvidos e não resolvidos.
<input type="checkbox"/> 5. Concordo totalmente <input type="checkbox"/> 3. Não sei opinar <input type="checkbox"/> 1. Discordo totalmente <input type="checkbox"/> 4. Concordo parcialmente <input type="checkbox"/> 2. Discordo parcialmente
Justificativa:
Área chave: Avaliação
18. Utilizar técnicas de avaliação para avaliar produtos intermediários (documentos ou códigos), tais como requisitos funcionais para descobrir defeitos. As técnicas utilizadas são: Walk-throughs ou Revisões técnicas ou inspeções.
<input type="checkbox"/> 5. Concordo totalmente <input type="checkbox"/> 3. Não sei opinar <input type="checkbox"/> 1. Discordo totalmente <input type="checkbox"/> 4. Concordo parcialmente <input type="checkbox"/> 2. Discordo parcialmente
Justificativa:
Área chave: Testes de baixo nível
19. Conscientizar os desenvolvedores sobre a importância do teste unitário para garantir a qualidade do software. O teste unitário deve ser considerado como um investimento e como um retorno positivo e rápido aumentando a confiança na equipe. Testar o sistema antes de passar para equipe de testes torna o trabalho mais simples e motivador. A equipe produzirá mais e terá tempo para explorar outras partes do sistema a fim de encontrar inconformidades e entregar um sistema com maior qualidade.
<input type="checkbox"/> 5. Concordo totalmente <input type="checkbox"/> 3. Não sei opinar <input type="checkbox"/> 1. Discordo totalmente <input type="checkbox"/> 4. Concordo parcialmente <input type="checkbox"/> 2. Discordo parcialmente
Justificativa: