Instituto de Pesquisas Tecnológicas do Estado de São Paulo

Alessandra Ap. Sericolla Diniz

Processo de Inspeção em Artefatos de Testes Funcionais de Software

São Paulo
2008

Alessandra Ap. Sericolla Diniz

Processo de Inspeção em Artefatos de Testes Funcionais de Software

Dissertação de Mestrado apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo-IPT, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Computação
Área de Concentração: Engenharia de Software

Orientador: Prof. Dr. Edison Spina

São Paulo
2008

Alessandra Ap. Sericolla Diniz

Processo de Inspeção em Artefatos de Testes Funcionais de Software

Dissertação de Mestrado apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo-IPT, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Computação
Área de Concentração: Engenharia de Software

Data da aprovação: 30/10/2008

_____
Prof. Dr. Edison Spina (Orientador)
Escola Politécnica da Universidade de São Paulo

Membros da Banca Examinadora:

Prof. Dr. Edison Spina (Orientador)
Escola Politécnica da Universidade de São Paulo

Prof. Dr. Marco Túlio Carvalho de Andrade (Membro)
Escola Politécnica da Universidade de São Paulo

Prof. Dr. José Eduardo Zindel Deboni (Membro)
IPT – Instituto de Pesquisas Tecnológicas do Estado de São Paulo

Agradecimentos

Agradeço a Deus por ter me dado vida, repleta de graça e misericórdia, e que me permitiu concluir mais uma etapa em minha jornada.

Ao meu esposo Luiz Fernando pelo incentivo, paciência, compreensão e, principalmente pelo amor que tem me dado nestes últimos doze anos.

Aos meus filhos Leonardo e Isabela, que são a luz da minha vida e razão do meu viver.

Aos meus pais Ana e Luiz que sempre me apoiaram e incentivaram meus estudos.

Ao Prof. Dr. Edison Spina, pelo apoio e dedicação durante a orientação desta dissertação e pela confiança em mim depositada.

Ao Prof. Dr. José Eduardo Deboni e Prof. Dr. Marco Túlio Andrade participantes da banca e que muito contribuíram para a evolução deste trabalho.

A todos os meus amigos que de forma direta ou indireta contribuíram para a conclusão deste trabalho.

SUMÁRIO

Lista de Tabelas

Lista de Ilustrações

Lista de Abreviaturas e Siglas

CMM	*Capability Maturity Model*
CMMI	*Capability Maturity Model Integration*
DBR	*Defect-Based Reading*
GQS	Garantia de Qualidade de Software
GQM	*Goal/Question/ Metric*
IEEE	*Institute of Electrical and Electronics Engineers*
ISO	*International Organization for Standardization*
OpenUP	*Open Unified Process*
PBR	*Perspective-Based Reading*
RUP	*Rational Unified Process*
SPICE	*Software Process Improvement and Capability dEtermination*
SQA	*Software Quality Assurance*
OORTs	*Object-Oriented Reading Techniques*
UBR	*Usage-Based Reading*
UP	*Unified Process*
VV&T	Verificação, Validação e Testes

Resumo

Esta dissertação foi uma proposta de um processo de inspeção em artefatos gerados durante o processo de teste de software. Para que o processo de teste de software seja desenvolvido corretamente, é necessário que envolva planejamento, execução dos testes e avaliação dos resultados. Cada uma destas atividades deve usar padrões pré-estabelecidos e gerar artefatos de saída. O trabalho descreve a especificação detalhada de um processo de inspeção desses artefatos gerados durante a fase de testes funcionais de software. Foram criadas listas de verificação para apoiar o processo de inspeção e, para a avaliação da proposta, foi realizado um comparativo com as listas de verificação dos processos RUP e OpenUP, além de um estudo de caso realizado para a validação do processo de inspeção. A aplicação dos processos propostos permitiu observar a melhora na qualidade dos artefatos de teste desenvolvidos.

Palavras-chaves: Inspeção de Software, Testes de Software, Lista de Verificação, Artefatos de Teste, Qualidade de Software.

Abstract

This master thesis was a proposal for an inspection process in artifacts generated during the software testing process. In order to ensure a properly developed software testing process, it is necessary planning, implementation and test results evaluation activities. Each of these activities should use pre-established patterns and will generate output artifacts. This paper describes a detailed specification of an inspection process in artifacts generated by software functional testing process. Checklists were created to support the inspection process and proposal evaluation, and a comparison was made with RUP and OpenUP checklists, as well as a case study was conducted to process validation. The implementation of process proposed allowed observing the improvement in quality on test artifacts developed.

Keywords: Software Inspection, Software Testing, Checklist, Test Artifacts, Software Quality.

1 INTRODUÇÃO

Os negócios dependem cada vez mais do uso de programas de computador para garantirem a sobrevivência das organizações nos mercados concorridos. Esta necessidade exige que o software seja desenvolvido com qualidade, dentro dos prazos e custos estimados, tenha baixos índices de defeitos e certa flexibilidade que permita mudanças rápidas e constantes.

O resultado da qualidade final do produto de software está diretamente relacionado à qualidade do processo de desenvolvimento de software. Existem diversos modelos e normas, dentre elas ISO/IEC 15504 SPICE (*Software Process Improvement and Capability dEtermination),* CMM (*Capability Maturity Model)* e CMMI (*Capability Maturity Model Integration),* que procuram tornar o processo de software mais eficiente (ROCHA, 2001).

A função da Garantia de Qualidade de Software (GQS), ou *SQA* (*Software Quality Assurance*), é proposta como parte dos processos de melhoria de software e diversos estudos (PAULK, 1999; CMMI, 2006; PRESSMAN, 2001) descrevem seus objetivos: o controle e a garantia da qualidade, avaliação da conformidade das atividades reais com as práticas adotadas, e a coleta de métricas para a reavaliação dos processos e estimativas. As atividades de verificação, validação e testes (VV&T) fazem parte do plano de garantia de qualidade, e os processos de revisão e teste complementam-se para atingir os objetivos da GQS (PARNAS, 2003).

De acordo com (PRESSMAN, 2001) as revisões são atividades de verificação, utilizadas pelo processo de GQS, e devem ser executadas por um grupo de pessoas que examinam produtos de trabalho, dentre eles, especificação de requisitos, código fonte ou qualquer outro artefato produzido, e identificam defeitos que podem ser removidos antes que o artefato seja enviado para as etapas seguintes do processo de desenvolvimento.

O processo de inspeção foi proposto inicialmente por Fagan (1976) e caracteriza-se por ser um tipo de revisão que pode ser aplicada a todos os artefatos gerados durante o processo de desenvolvimento de um software, constando de um processo de detecção de defeitos rigoroso e bem definido. Diversas técnicas de inspeção têm sido propostas desde então (BASILI, 1997; PORTER, 1995; SHULL, 1998; TRAVASSOS, 1999; THELIN,

2004), porém, o enfoque destas técnicas normalmente é voltado para a inspeção de código-fonte, documentos de requisitos e documentos de projeto.

O teste de software é uma das atividades de validação e verificação que compõe a GQS, e o conceito de teste, segundo Myers, é o processo de executar um programa com a intenção de encontrar erros (MYERS, 2004), além de validar os requisitos definidos pelos usuários. O processo de teste envolve as etapas de planejamento, projeto de casos de teste, execução e avaliação dos resultados, e a adoção de um método de testes estruturado, baseado nos modelos e normas vigentes no mercado, auxilia a execução desta atividade (ROCHA, 2001).

No trabalho desenvolvido por Doria (2001) foi feito um estudo comparativo, utilizando a replicação de experimentos, entre técnicas de teste e técnicas de inspeção no que se refere à detecção de defeitos em produtos de software, mais especificamente em código fonte e documento de especificação de requisitos.

No trabalho de Ciolkowski (2002), foram realizados experimentos com diversos tipos de inspeções, inicialmente em ambiente de pesquisa e em seguida foi analisada a influência das inspeções na prática da indústria de software. Ciolkowski (2002) sugere ainda que sejam desenvolvidos novos trabalhos que melhor integrem as inspeções dentro do processo de desenvolvimento de software, relacionando-as com outras técnicas de detecção de defeitos tais como testes.

O estudo desenvolvido por Winkler (2005) apresenta a execução de experimentos com inspeções baseadas em técnicas de leitura, integrando cenários de teste e técnicas de inspeção.

Para o desenvolvimento deste trabalho o termo processo foi utilizado com o objetivo de determinar de forma clara e detalhada quem faz o que, quando e como. Um processo é composto por atividades, que devem ser realizadas por um ou mais membros de uma equipe de trabalho. As atividades geradas dentro de um processo produzem como resultados artefatos de saída, e podem requerer artefatos de entrada para o seu desenvolvimento. Artefatos são documentos, modelos ou códigos gerados dentro de uma atividade, e que posteriormente poderão ser inspecionados. O termo artefato utilizado neste

trabalho refere-se aos documentos gerados durante os processos de teste e inspeção de software.

1.1 Objetivos

O objetivo deste trabalho foi desenvolver um processo de inspeção em artefatos de teste de software, amparado por um processo de testes funcionais de software. O processo de testes foi desenvolvido utilizando a perspectiva de qualidade do usuário, na qual se espera que os requisitos levantados junto aos usuários sejam plenamente atingidos pelo produto que está sendo entregue. O trabalho foi realizado utilizando a técnica de testes funcionais, próprias de empresas que utilizam o modelo de fornecimento por fábrica de software, pois, nestes casos, os testes estruturais são de responsabilidade da fábrica que desenvolve o código.

1.2 Resultados Esperados e Contribuições

Durante o desenvolvimento do trabalho foram propostos artefatos que apóiam o processo de testes (Plano de Teste, Especificação de Caso de Teste, Especificação de Procedimento de Teste, Relatório de Incidente de Testes e Relatório de Resumo de Testes), baseados em normas e modelos existentes, e que permitem a execução do processo de inspeção nos artefatos gerados durante a execução dos testes. Foram propostos também artefatos de apoio ao processo de inspeção (Listas de Verificação e Registro de Inspeção).

Espera-se contribuir com melhorias nas técnicas de inspeção existentes, agregando aos processos atuais, a inspeção em artefatos de teste de software, pois a bibliografia analisada dá ênfase principalmente aos artefatos gerados durante a especificação de requisitos e ao código-fonte dos programas.

1.3 Método de Trabalho

Para atingir os objetivos propostos no trabalho foram executadas as seguintes atividades:

- **Revisão**: revisão bibliográfica sobre os temas envolvidos, que serviram de base para o desenvolvimento do trabalho. Foram analisados os seguintes tópicos:

  a) Garantia da qualidade de software;

  b) Atividades de verificação, validação e testes de software;

  c) Estratégias de inspeção de artefatos de software;

- **Análise bibliográfica**: identificação das atividades necessárias para a melhoria do processo de teste de software, dentro dos modelos e normas avaliados na bibliografia;

- **Processo de testes**: especificação do processo de testes a ser utilizado e definição de artefatos necessários para a execução dos testes, que foram utilizados como entrada para o processo de inspeção em artefatos de testes;

- **Inspeção de testes:** especificação do processo de inspeção em artefatos de testes, definindo suas entradas, saídas e os passos de execução, para efetivamente resultar num nível de qualidade que satisfaça a expectativas dos clientes. Fazem parte do escopo deste trabalho somente os testes funcionais de software;

- **Avaliação do Processo de Inspeção:** avaliação das listas de verificação propostas através de comparativo com processos de engenharia de software existentes. Realização de estudo de caso e documentação dos resultados obtidos. Foram selecionados alguns projetos que envolveram desenvolvimento de software, em que foram desenvolvidas as especificações de requisitos, necessárias para a realização da inspeção dos artefatos de testes. Avaliação dos resultados obtidos e análise da eficácia dos processos de testes e de inspeção.

1.4 Organização do Trabalho

Este trabalho é composto por seis capítulos e um apêndice.

O capítulo 1, Introdução, apresenta a motivação para a realização do trabalho, os objetivos, a metodologia desenvolvida e a descrição da estrutura do trabalho.

O capítulo 2, Revisão Bibliográfica, apresenta o resultado da pesquisa bibliográfica realizada, formando o embasamento para o desenvolvimento do trabalho. O capítulo possui os seguintes temas principais: descrição do processo de garantia da qualidade de software e descrição das atividades de verificação, validação e testes. O capítulo apresenta também os principais estudos envolvendo inspeção de artefatos de software.

O capítulo 3, Processo de Testes, apresenta os objetivos a serem atingidos com a implantação do processo de testes e os requisitos necessários para a sua implementação. Apresenta também a especificação do processo e a descrição dos artefatos de testes necessários, que compõem o processo de inspeção em artefatos de testes.

O capítulo 4, Processo de Inspeção de Testes Funcionais, apresenta a definição dos passos necessários para a execução da inspeção nos artefatos de testes funcionais, especificando suas entradas, saídas, o conjunto de atividades que compõe o processo, além de identificar os responsáveis pela execução. São apresentados também os requisitos de qualidade usados para criação das listas de verificação e os critérios de avaliação dos artefatos inspecionados.

O capítulo 5, Avaliação do Processo de Inspeção, apresenta a aplicação do processo de teste proposto e da inspeção nos artefatos de testes, e os resultados obtidos com o processo.

O capítulo 6, Considerações Finais, apresenta a avaliação da realização deste trabalho, suas limitações, bem como propostas para o seu desenvolvimento no futuro.

O Apêndice A apresenta os documentos propostos que compõe o processo de testes.

2 REVISÃO BIBLIOGRÁFICA

Na tentativa de melhorar os processos de desenvolvimento de software, diversos modelos, normas e padrões foram criados por organizações internacionais, com o objetivo de aumentar a qualidade do produto gerado. As principais iniciativas na criação de modelos para avaliação e melhoria de processo de software são: os modelos CMM (*Capability Maturity Model*) e CMMI (*Capability Maturity Model Integration*) e a norma ISO/IEC 15504 (SPICE).

Em cada um destes modelos e normas, diversas atividades são propostas para alcançar o objetivo de qualidade em um software, e, em todas, é fundamental o papel da Garantia da Qualidade de Software (GQS).

O processo de Garantia de Qualidade de Software é composto por atividades de verificação, validação e testes (VV&T). O objetivo destas atividades é garantir que os procedimentos e padrões definidos para assegurar a qualidade do produto final estejam sendo seguidos. Neste capítulo é apresentada a revisão bibliográfica das principais técnicas de revisão e teste, além de modelos e normas que são aplicados nos processos de teste e revisão.

O IEEE (*Institute of Electrical and Eletronics Engineers)* apresenta as diferenças entre os termos defeito, engano, falha e erro. Defeito (*fault*) é um passo, processo ou definição de dados incorretos e engano (*mistake*) é a ação humana que produz um defeito, e os dois referem-se a conceitos estáticos (IEEE,1990). Um defeito existente pode resultar na ocorrência de um erro (*error*) durante a execução de um programa e a diferença entre o resultado esperado e o real pode levar a uma falha (*failure*) (IEEE, 1990). Para o desenvolvimento deste trabalho, durante a execução do processo de inspeção, o termo defeito será utilizado, pois se refere à busca de instruções incorretas em documentos, através de um processo estático.

2.1 CMM – *Capability Maturity Model*

O CMM (*Capability Maturity Model*) foi criado pelo *Software Engineering Institute* (SEI) da *Carnegie Mellon University*, a pedido do Departamento de Defesa dos Estados Unidos com o objetivo de avaliar a capacidade de potenciais fornecedores de software (PAULK, 1999).

O CMM descreve elementos chaves para melhoria e avaliação do processo de software através de um roteiro seqüencial e pode ser classificado como um modelo de melhoria de processo por estágios. Os modelos por estágios focam a maturidade organizacional através de um caminho evolutivo para a melhoria do processo. As áreas do processo são agrupadas em níveis de maturidade, que devem ser atendidas na sua totalidade para viabilizar um estágio definido de melhorias.

Na estrutura do modelo CMM cada nível define o grau de maturidade dos processos de uma organização e é composto por várias áreas-chave de processo. Cada área chave permite alcançar um conjunto de metas ou objetivos. A satisfação desses objetivos é que permite dizer se a organização atingiu um determinado nível de maturidade (PAULK, 1999). A Figura 1 apresenta os cinco níveis de maturidade do modelo.

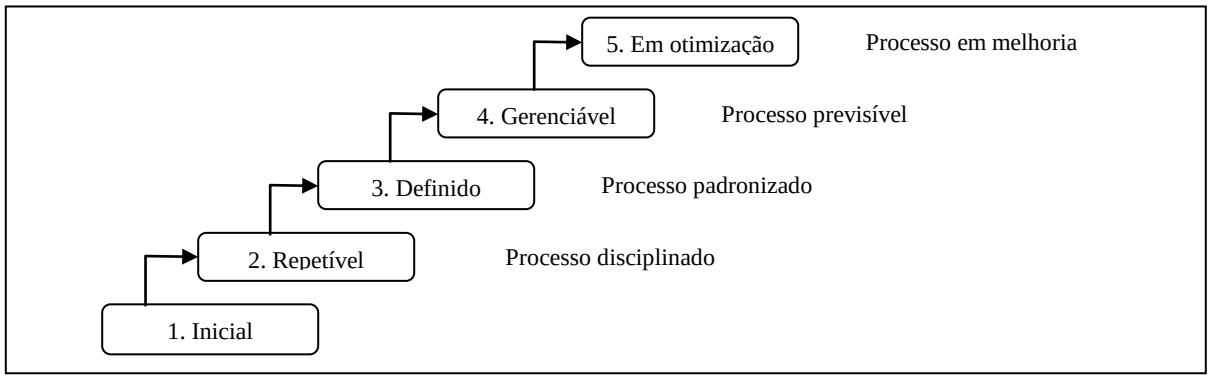

Figura 1 - Níveis de Maturidade do CMM (PAULK, 1999)

Baseados na estrutura geral do CMM foram criados alguns modelos específicos para atender diversas áreas e aspectos de uma organização, tais como a área de recursos humanos (*People* CMM), de sistemas em geral (SE-CMM), de software (SW-CMM), entre outras. O SW-CMM abrange práticas de planejamento, engenharia e gerência para o

desenvolvimento e manutenção de software, auxiliando uma organização a transformar seus processos de software imaturos em processos maduros e disciplinados.

O teste sistemático é uma atividade essencial para a ascensão ao nível 3 do modelo CMM, além de ser fundamental para as atividades de depuração, manutenção e estimativa de confiabilidade de software, devendo ser planejada a sua execução ao longo de todo o processo de desenvolvimento.

2.2 CMMI (*Capability Maturity Model Integration*)

O modelo CMMI (*Capability Maturity Model Integration*) é uma evolução do SW-CMM (AHERN, 2001). No CMMI foi definida uma área de gerência de projeto composta por seis áreas de processo: planejamento de projeto, acompanhamento e controle de projeto, gerenciamento de acordos com fornecedores, gerenciamento de integrado de projeto, gerenciamento de risco, e gerenciamento quantitativo de projeto.

O modelo CMMI teve a sua versão 1.1 lançada em 2002. Essa versão foi baseada nos rascunhos da versão 2.0 do CMM. O CMMI define duas representações de modelos de maturidade: em estágios e contínua. A representação em estágios é similar ao modelo de maturidade definido pelo CMM. A representação contínua, como a Norma ISO/IEC 15504, introduz o conceito de nível de capacidade de desenvolvimento, que difere do conceito de nível de maturidade usado pela representação em estágios por focar de forma individual nas áreas chave de processo, ao invés de lidar com o processo organizacional como um todo.

A representação contínua do CMMI é composta por seis níveis de capacidade: (0) Incompleto; (1) Executável; (2) Gerenciável; (3) Definido; (4) Quantitativamente Gerenciável; e (5) Otimizado. Em cada um desses níveis de capacidade, a partir do nível 1, existem objetivos genéricos associados. Em cada objetivo genérico foram definidas práticas genéricas que devem ser atendidas, para que a área do processo cumpra o objetivo genérico e possa ser classificada no nível de capacidade que esse objetivo genérico está relacionado (AHERN, 2001).

2.3 Norma ISO / IEC 15504 - SPICE

A norma ISO/IEC 15504 teve sua publicação mais recente em 2003, sendo desenvolvida através do projeto *SPICE* (*Software Process Improvement and Capability dEtermination)* e foi originada a partir do surgimento de diversos modelos de qualidade específicos para o desenvolvimento de software, gerando a necessidade de um padrão internacional (ISO, 2003).

A norma ISO/IEC 15504 foca a capacidade de uma parte do processo e oferece um caminho mais flexível para a implementação de melhorias. Ela permite que as organizações escolham áreas específicas do processo de software para a implementação de melhorias, bem como implementem níveis diferentes de capacidade para diferentes processos.

O SPICE é estruturado em um modelo bidimensional: a dimensão de processos define os processos a serem avaliados, identificando os pontos fortes e fracos que serão utilizados para um plano de melhoria, e a dimensão de capacidade define um modelo de medição, que permite determinar a capacidade de um processo para atingir seus objetivos.

Na dimensão de capacidade de processos, há uma classificação em seis níveis, que representam a evolução da capacidade dos processos implementados. Para cada nível está relacionado um conjunto de atributos de processo que devem ser atendidos. A Tabela 1 mostra os níveis de capacidade definidos no modelo SPICE e o atributo a ele relacionado.

Níveis de Capacidade	Atributos de Processo
Nível 0: Processo Incompleto	---
Nível 1: Processo Executado	Execução de processo
Nível 2: Processo Gerenciado	Gestão de execução Gestão de produto de trabalho
Nível 3: Processo Estabelecido	Definição de processo Recursos de processo
Nível 4: Processo Previsível	Definição de processo Recursos de processo
Nível 5: Processo em Otimização	Mudança de processo Melhoria contínua

Tabela 1 - Níveis de capacidade e seus atributos

Na dimensão de processos, cinco categorias agrupam os processos em relação ao tipo de atividade que executam. As categorias são: cliente-fornecedor, engenharia, suporte, gerência e organização. Cada processo possui: Identificação, Nome, Propósito, Resultados, um conjunto de Práticas Base e determinados Produtos de Trabalho. Os Produtos de Trabalho de um processo são aqueles utilizados e/ou produzidos pela execução do processo.

As Práticas Base são atividades que ao serem executadas contribuem para o atendimento do propósito de um processo. Para cada Prática Base estão relacionados os resultados que a prática ajuda a alcançar. Para o desenvolvimento do trabalho foram consideradas as Práticas Básicas que envolvem as atividades de verificação, validação e teste de software, conforme apresentado na Tabela 2.

Processo	Descrição
CUS - Cliente-Fornecedor	Processos que impactam diretamente os produtos e serviços de software do fornecedor para o cliente.
CUS.2	Gerenciar necessidades do Cliente
ENG - Engenharia	Processos que especificam, implementam ou mantém um sistema ou produto de software e sua documentação.
ENG.3	Desenvolver o projeto do software
ENG.5	Integrar e testar o software
ENG.6	Integrar e testar o sistema
SUP - Suporte	Processos que podem ser empregados por outros processos.
SUP.1	Desenvolver a documentação
SUP.2	Desempenhar a gerência de configuração
SUP.3	Executar a garantia da qualidade
SUP.4	Executar a verificação dos produtos de trabalho
SUP.5	Executar a validação dos produtos de trabalho
SUP.6	Executar revisões conjuntas
SUP.7	Executar auditorias
MAN – Gerência	Processos que contém práticas de natureza genérica que podem ser usadas por quem gerencia projetos ou processos dentro de um ciclo de vida de software.
MAN.2	Gerenciar a qualidade

Tabela 2 - Práticas Básicas que envolvem atividades de VV&T

2.4 Técnicas de Revisão

Durante as etapas de desenvolvimento de um software diversos tipos de artefatos são produzidos como saídas em cada uma das etapas. Garantir a qualidade dos artefatos produzidos em todas as etapas intermediárias é importante para o resultado do processo final. Dentro das atividades de verificação do processo de Garantia de Qualidade de Software, as revisões são utilizadas com o objetivo de encontrar defeitos durante o desenvolvimento de uma etapa e evitar que eles propaguem para as etapas seguintes do processo de desenvolvimento (LAITENBERGER, 2000).

De acordo com o padrão IEEE 1028 (IEEE, 1997) a revisão de software visa avaliar produtos de software por uma equipe tecnicamente qualificada com o intuito de identificar discrepâncias do produto, com relação aos padrões e especificações pré-definidos.

Segundo Pressman (2001) as revisões servem como um filtro para os engenheiros de software, sendo aplicadas em diversos pontos durante o ciclo de vida do desenvolvimento do software, podendo ser classificadas como revisões formais ou informais.

As revisões informais são normalmente realizadas pelos grupos de desenvolvimento, não envolvem preparação e registro de informações e não realizam a coleta de métricas. Em geral as revisões informais discutem aspectos técnicos do software sem a preocupação de seguir padrões pré-estabelecidos para a realização das revisões (COLLOFELLO, 1988).

As revisões formais seguem procedimentos formais previamente descritos e é obrigatória a realização da etapa de preparação antes da reunião de inspeção. Para classificar uma revisão como formal é necessário que existam de padrões, procedimentos, regras para criação de artefatos, listas de verificação e formulários a serem utilizados durante a revisão, previamente definidos (COLLOFELLO, 1988). As revisões formais permitem a coleta de métricas para a posterior análise de dados com o objetivo de melhoria contínua do processo.

As revisões formais envolvem um grupo de pessoas que examina parte ou todo o processo de software, o sistema ou a sua documentação associada, a fim de descobrir

possíveis problemas. As conclusões das revisões devem ser formalmente registradas e passadas para o autor ou para o responsável em corrigir os problemas constatados.

Segundo Pressman (2001) os objetivos das revisões técnicas formais são:

- Descobrir defeitos de função, lógica ou implementação em qualquer artefato de software;
- Verificar se o software atende aos seus requisitos;
- Garantir que o software tenha seguido os padrões pré-definidos;
- Garantir que o software tenha sido desenvolvido de maneira uniforme;
- Melhorar o gerenciamento dos projetos.

A Tabela 3 descreve resumidamente os principais tipos de revisão.

Tipos de Revisão	Características
Inspeção de Projeto ou Programa	A revisão deve ser orientada por uma lista de verificação (*checklist)* de possíveis defeitos, envolvendo a leitura, passo a passo de um determinado produto de software.
Walkthroughs	Assim como a inspeção, a revisão *walkthrough* é um método de revisão formal baseada na leitura de um documento de software, a diferença está na maneira como a reunião deve ser conduzida. Os participantes devem, além de ler o artefato e checar os defeitos através de uma lista de verificação (*checklist)*, simular a execução do programa.
Revisão por Pares – *Peer Review*	É uma técnica de revisão realizada no código de programa e deve ser feita por pares de programadores. Os programadores devem ter o mesmo nível de conhecimento e preferencialmente pertencer a projetos diferentes. A finalidade é conseguir benefícios através da exposição de pontos de vista diferentes.

Tabela 3 - Tipos de Revisão

2.4.1 Inspeção de Artefatos de Software

O modelo de revisão por inspeção foi proposto em (FAGAN, 1976) e caracteriza-se por ser um processo rigoroso e bem definido, que pode ser aplicado a diversos tipos de artefatos de software, tais como, especificação de requisitos, projetos, código e plano de testes.

A inspeção de artefatos de software pode ser realizada em todas as fases de desenvolvimento do projeto. Seus objetivos são a detecção de defeitos no software, além da prevenção, detectando os defeitos antes que eles sejam enviados para novas fases do processo de desenvolvimento, pois a descoberta de um determinado defeito em uma fase posterior acarreta um custo maior para a sua solução.

Segundo Boehm (2001) a inspeção aplicada em artefatos de software visando à melhoria da qualidade permite identificar até 60% dos defeitos contidos nesses artefatos. E para Shull (2000 apud DENGER, 2007) o percentual de defeitos identificados em uma inspeção pode alcançar entre 60 a 90% dos defeitos existentes no artefato analisado.

Em um processo de inspeção, as pessoas envolvidas devem verificar se um produto de trabalho satisfaz as especificações do produto de trabalho antecessor, como, por exemplo, documento de requisitos e de projeto; devem também identificar se existem desvios dos padrões pré-estabelecidos, além de sugerir melhorias para o autor e promover troca de experiências entre os participantes.

Durante a execução da inspeção, os participantes recebem um ou mais papéis e responsabilidades. Na Tabela 4 são descritos os papéis dos participantes de uma reunião de inspeção.

De acordo com Porter (1995) existem três tipos de inspeção:

- *Ad-Hoc*: não possuem um procedimento sistemático e são baseadas na experiência do inspetor.
- *Checklists* ou Lista de Verificações*:* são utilizados questionários baseados em lições aprendidas em inspeções anteriores, permitindo identificar os defeitos comuns mais rapidamente.
- Técnicas de Leitura: são procedimentos sistemáticos e bem definidos para inspeção de um artefato de software.

Papéis	Atividades
Autor	É o criador ou mantenedor do produto de trabalho a ser inspecionado. É responsável por entregar o artefato a ser inspecionado e os documentos relacionados aos demais participantes, além de esclarecer dúvidas sobre estes artefatos.
Moderador	Tem o papel de Facilitador da Inspeção, planeja o cronograma e lidera a inspeção. Identifica os demais participantes do processo de inspeção, determinando o status do produto de trabalho ao final.
Leitor	Faz a leitura de partes no produto de trabalho inspecionado, de maneira a fazer com que o time de inspeção apresente comentários, não-conformidades ou questionamentos.
Escritor	Registra e classifica as não-conformidades encontradas durante a inspeção.
Inspetor	Deve examinar o produto de trabalho antes da reunião de inspeção para encontrar defeitos e desvios. Participa da reunião de inspeção para identificar defeitos, desvios e sugerir melhorias.
Coordenador das Inspeções	É responsável pelas métricas de inspeção do projeto, além de manter os registros das inspeções conduzidas e dados do sumário de cada inspeção.

Tabela 4 - Papéis dos Participantes em uma Reunião de Inspeção

2.4.2 Técnicas de Leitura para Artefatos de Software

As técnicas de leitura foram desenvolvidas para aumentar o desempenho da inspeção de software, quanto à atividade de detecção de defeitos. Uma técnica de leitura pode ser caracterizada como uma série de passos para a análise individual de um artefato de software de forma a permitir a extração do entendimento necessário para a execução de uma determinada tarefa (BASILI, 1996).

Dentre as técnicas de leitura de artefatos descritas na literatura, pode-se destacar: leitura baseada em defeitos, leitura baseada em perspectivas, técnicas de leitura para orientação a objeto e leitura baseada em casos de uso.

2.4.2.1 Leitura Baseada em Defeitos - DBR (*Defect-Based Reading*)

A leitura baseada em defeitos (DBR) representa uma família de técnicas de leitura para a detecção de defeitos em documentos de requisitos que utiliza diagramas de máquina de estado (CIOLKOWSKI, 2002).

Na leitura baseada em defeitos, para cada classe de defeitos, um grupo de questões é desenvolvido para caracterizar a classe de defeitos, por exemplo, a classe de tipo de dados inconsistentes. As questões também apresentam os cenários, que são os passos que deverão ser seguidos durante a leitura do documento. Enquanto o documento é lido, seguindo os passos pré-definidos, o revisor tenta responder as questões que compõe o cenário.

2.4.2.2 Leitura Baseada em Perspectiva - PBR (*Perspective-Based Reading*)

A leitura baseada em perspectiva (PBR) representa uma família de técnicas de leitura elaborada inicialmente para a detecção de defeitos em documentos de requisitos escritos em linguagem natural (SHULL, 1998). Posteriormente teve sua utilização estendida para inspeções de projeto e código-fonte.

A PBR auxilia a detecção de defeitos analisando quais informações nos requisitos possuem maior relevância nas diferentes formas de utilização do documento. Por exemplo, em um modelo de ciclo de vida de software, um documento de requisitos pode ser utilizado para a construção do modelo de projeto, para a elaboração do plano de teste do sistema e para descrição das funcionalidades do sistema.

Desta forma, a leitura baseada em perspectiva permite que o revisor avalie o documento segundo uma dessas perspectivas fornecendo um procedimento para orientá-lo a criar um modelo físico baseado nos requisitos. A construção do modelo ajuda o revisor a focalizar sua revisão. Sendo assim, o revisor que assume a perspectiva de testador constrói um conjunto de casos de testes, o revisor responsável pela perspectiva de projetista constrói um modelo de projeto e o revisor responsável pela perspectiva de usuário constrói um manual contendo as funcionalidades do sistema.

2.4.2.3 Técnicas de Leitura Orientada a Objetos - OORTs (*Object-Oriented Reading Techniques*)

As técnicas de leitura orientada a objetos (OORTs) representam uma família de sete técnicas de leitura que indicam um procedimento para as revisões individuais dos diferentes diagramas e documentos de projetos orientados a objeto, construídos com a notação UML (TRAVASSOS, 1999). O processo de leitura utilizando OORTs é realizado em duas dimensões de leitura, horizontal e vertical. Diversos diagramas de projeto são verificados na leitura horizontal, para assegurar que estejam consistentes entre si. Na leitura vertical os modelos de projeto são validados para garantir que o projeto do sistema está correto com relação aos requisitos.

2.4.2.4 Leitura Baseada em Casos de Uso - UBR (*Usage-Based Reading*)

A UBR é uma técnica de leitura que auxilia a detecção de defeitos severos do ponto de vista do usuário (THELIN, 2004) e foca na leitura orientada por um modelo de casos de uso priorizados em ordem de importância para o usuário do sistema, durante a fase de preparação de um processo de inspeção de software. A idéia da UBR é permitir que as expectativas do usuário governem a inspeção, utilizando os modelos de caso de uso nas inspeções realizadas em todas as fases de desenvolvimento do software.

Durante a inspeção, os revisores lêem o documento executando manualmente os casos de uso e tentando detectar defeitos que são mais importantes de acordo com a prioridade estabelecida para o usuário. Segundo os autores (THELIN, 2003), a UBR está relacionada com a perspectiva de usuário de PBR, porém, os autores da UBR apontam algumas diferenças (THELIN, 2003):

- Na UBR os casos de uso servem como guias para os artefatos inspecionados e na perspectiva de usuário de PBR os revisores desenvolvem casos de uso baseados no documento de requisitos, de modo a detectarem defeitos.
- O objetivo de UBR é aumentar a eficiência e a efetividade das inspeções focando os casos de uso mais importantes sob o ponto de vista dos usuários

e na PBR o objetivo é melhorar a efetividade da inspeção minimizando a sobreposição entre os defeitos encontrados por diferentes revisores;

- Os cenários UBR são específicos do projeto que está sendo inspecionado, enquanto os cenários de PBR são genéricos, pois, os cenários desenvolvidos para um tipo de artefato, em um determinado projeto, poderão ser reutilizados em outro projeto.

2.5 Técnicas de Teste de Software

O teste de software é uma atividade de verificação e validação que envolve a execução dinâmica do software com o objetivo de encontrar defeitos e verificar se este atende aos requisitos propostos.

Myers destaca os principais objetivos das atividades de teste: (MYERS, 2004)

- Testar é o processo de executar um programa com a intenção de encontrar erros;
- Um bom caso de teste é aquele com alta probabilidade de descobrir um erro ainda não identificado;
- Um teste bem sucedido é aquele que descobre um erro ainda não identificado.

A atividade de teste é composta de quatro etapas: planejamento, projeto de casos de teste, execução e avaliação dos resultados. Estas etapas são realizadas durante todo o processo de desenvolvimento do software (ROCHA, 2001).

As técnicas de teste são classificadas pela origem das informações e estão resumidas na Tabela 5 (PRESSMAN, 2001; MYERS, 2004).

Para o desenvolvimento do trabalho foi considerada a técnica de teste funcional, verificando se o processo de teste desenvolvido consegue alcançar os requisitos propostos para o software.

Técnicas de Teste	Características
Funcional ou Caixa-Preta	Os testes são analisados considerando as interfaces (entradas/saídas) do produto de software, para demonstrar se as funcionalidades e operações realizadas estão produzindo resultados corretos.
Estrutural ou Caixa-Branca	São os testes realizados em uma determinada implementação, verificando detalhes do código, executando partes ou componentes elementares do programa.

Tabela 5 - Técnicas de Testes

2.6 Norma IEEE 829 - *Standard for Software Test Documentation*

A norma IEEE 829 descreve uma lista de documentos a serem utilizados nas atividades de teste de software. Os oito documentos propostos na norma auxiliam na preparação, execução e registro das atividades de teste. Quatro documentos referem-se ao plano de testes e às especificações e os outros quatro definem os relatórios de testes. A norma define o propósito, a estrutura e o conteúdo de cada documento (IEEE, 1998a).

Os documentos referentes ao plano de testes e as especificações são:

- **Plano de Teste:** apresenta o planejamento das atividades de testes, recursos necessários, cronograma, abordagem dos testes e ambiente requerido para a execução dos testes;
- **Projeto de Especificação de Testes:** detalha as condições de teste e os resultados esperados, identifica as características e funcionalidade a serem testadas, além dos critérios de aprovação dos testes;
- **Especificação de Caso de Teste:** define os casos de teste, detalhando os dados de teste, a serem utilizados nas condições de teste, identificados no Projeto de Especificação de Testes;
- **Especificação de Procedimento de Testes:** detalha como executar cada teste, incluindo as pré-condições e os passos que devem necessariamente ser seguidos.

Os relatórios de testes são cobertos por quatro documentos:

- **Relatório de Transmissão de Item de Teste:** apresenta a progressão dos componentes de teste de software, de um estágio de teste para o seguinte;
- **Diário de Teste:** apresenta quais testes foram executados, por quem e em qual ordem, e quais testes passaram ou falharam;
- **Relatório de Incidente de Testes:** detalha, para os testes que falharam, o resultado atual comparando-o com o resultado esperado;
- **Relatório de Resumo de Testes:** apresenta de forma resumida o resultados dos testes realizados, a garantia da qualidade do esforço de teste, a qualidade do software final e a estatística derivada do Relatório de Incidentes de Testes. O documento final é usado para indicar se o software atingiu a sua finalidade, de acordo com os critérios de aceitação definidos pelos envolvidos no projeto.

A norma estabelece o formato dos documentos, no entanto, ela não estipula quais devem ser produzidos, flexibilizando os projetos para que se adaptem às necessidades, agrupando alguns documentos propostos, além de permitir a redução do conteúdo dos documentos.

2.7 Norma ISO/IEC 12119 - *Software Packages - Quality Requirements and Testing* – Avaliação de Pacotes de Software

A norma ISO/IEC 12119 (ISO, 1994) é aplicada aos pacotes de software como processadores de texto, planilhas eletrônicas, bancos de dados, softwares gráficos, programas para funções técnicas ou científicas ou programas utilitários, estabelecendo requisitos de qualidade e instruções de teste a serem realizadas em um pacote de software, com relação aos requisitos definidos.

Os requisitos de qualidade definidos na norma são: descrição do produto, documentação do usuário e programas e dados. A descrição do produto apresenta as principais propriedades do pacote. A documentação do usuário é um documento que deve

ser avaliado em relação à sua completude, correção, consistência, inteligibilidade, apresentação e organização, conforme requisitos apresentados na Tabela 6. Programas e dados referem-se aos requisitos que devem estar descritos para o correto funcionamento do produto. De acordo com a norma, um pacote de software é considerado em conformidade quando atende aos requisitos de qualidade definidos.

Item	Requisitos
Completude	Deve conter as informações necessárias para o uso do produto de software, tais como estabelecer as funções do pacote, procedimentos de instalação e os valores limites.
Correção	A informação apresentada deve estar correta e sem ambigüidade.
Consistência	Deve haver plena coerência entre a documentação e a descrição do produto. Cada termo deve ter um único significado.
Inteligibilidade	A documentação deve ser compreensível pela classe de usuários que desenvolve atividades com o produto, utilizando termos apropriados, exibições gráficas e explicações detalhadas.
Apresentação e Organização	Deve ser apresentada através de uma forma que facilite uma visão geral, através de índices e tabelas de conteúdo. Se o documento não está na forma impressa, deve haver indicação de como efetuar a impressão.

Tabela 6 - Requisitos de Qualidade para Documentação do Usuário

A Norma 12119 define também as instruções para teste, incluindo o teste das propriedades necessárias a todos os produtos de mesmo uso, e o teste das propriedades especificadas na descrição do produto. A Figura 2 representa a estrutura da Norma 12119.

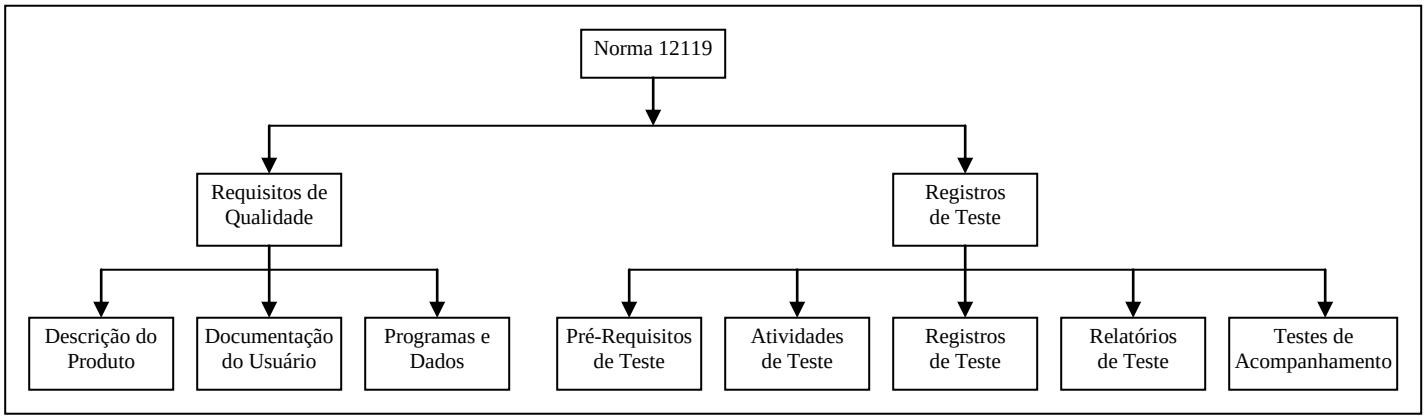

Figura 2 - Estrutura da Norma ISO/IEC 12119 (ISO, 1994)

2.8 Considerações Finais

Neste capítulo foram apresentados os principais conceitos relacionados Garantia de Qualidade de Software, aos processos de revisão de artefatos e de testes de software.

A filosofia apresentada nos modelos CMM, CMMI e na norma SPICE serviu como base para o desenvolvimento do trabalho, pois, apresentam uma seqüencia de passos pré-definidos a serem seguidos, baseados em uma documentação bem estruturada, e sugerem a implementação da função da GQS, como forma de avaliar o que está sendo feito durante o processo de desenvolvimento de software.

Foram apresentados os principais tipos de revisão e o detalhamento da atividade de inspeção e as técnicas praticadas para a sua realização.

As técnicas de teste foram explicadas dentro do contexto da dissertação.

Além disso, foram apresentadas: a norma IEEE 829, que regulamenta o padrão de documentação de teste de software, e a norma ISO/IEC 12119, que apresenta conceitos de avaliação de pacotes quanto a requisitos de qualidade e testes.

Estes conceitos formam a base para o processo de testes proposto na dissertação, que será apresentado no próximo capítulo.

3 PROCESSO DE TESTES

Neste capítulo são apresentados o processo de teste proposto e os objetivos a serem atingidos com a sua implantação. O capítulo apresenta também a especificação do processo, a definição de tarefas executadas e respectivas responsabilidades, procedimentos e a descrição dos artefatos de testes necessários, que irão compor o processo de inspeção em artefatos de testes. O processo de teste propostos foi derivado a partir da norma IEEE 829 (IEEE, 1998a).

Na Figura 3 é apresentada a visão geral dos processos de teste e de inspeção integrados. O processo de testes foi divido nas fases de planejamento, projeto, execução e avaliação dos resultados, e a proposta desta dissertação é a aplicação de rotinas de inspeção após a execução de cada uma destas atividades, criando uma retroalimentação a partir da integração destes processos. A retroalimentação permite que a avaliação dos resultados obtidos em uma determinada tarefa identifique pontos de melhorias que poderão ser aplicados durante uma nova execução desta mesma tarefa.

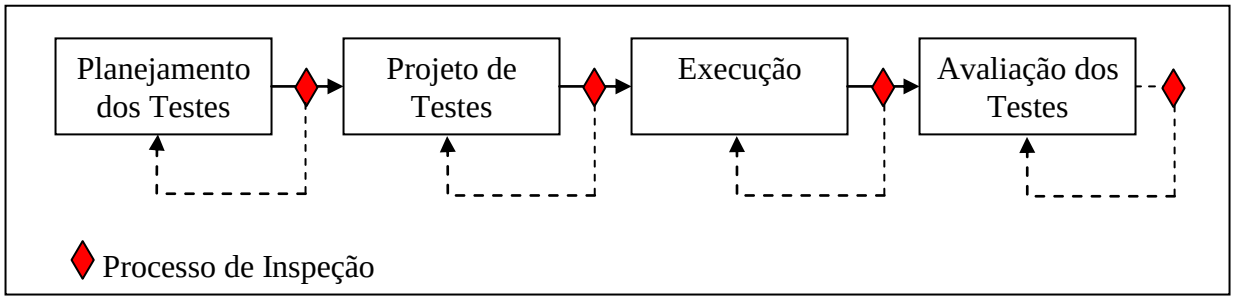

Figura 3 - Visão Geral dos Processos de Teste e Inspeção

A integração entre o processo de testes e o processo de inspeção gerou nova uma fase dentro do processo de testes: a fase de correção. A fase de correção dentro do processo de testes permite que os artefatos produzidos nas demais fases do processo de teste (planejamento, projeto, execução e avaliação) possam ser corrigidos o quanto antes, evitando que os defeitos existentes se propaguem para as demais fases do processo de desenvolvimento do software.

3.1 Objetivos do Processo de Teste

Os objetivos do processo de teste propostos são:

- Executar o software para descobrir erros ainda não detectados;
- Projetar casos de teste com elevada probabilidade de revelar erros;
- Projetar testes que descubram diferentes classes de erros com uma quantidade de tempo, esforços e custos menores;
- Verificar se os requisitos foram corretamente implementados;
- Proporcionar boa indicação da qualidade e confiabilidade do sistema.

3.2 Escopo

O escopo do processo de teste proposto é a definição dos procedimentos e métodos necessários para a execução dos testes funcionais de um projeto de software, que são aqueles utilizados para assegurar que o comportamento do sistema atende às especificações dos requisitos do sistema, e utiliza as entradas e saídas produzidas para avaliar seu comportamento (PRESSMAN, 2001).

Os testes funcionais são conhecidos como testes de caixa-preta, isto é, o conjunto de técnicas que considera o sistema como uma caixa fechada cuja estrutura interna não se conhece (MYERS, 2004). O testador somente conhece as entradas e os resultados esperados. Portanto, o testador estabelece sua estratégia de testes com base na especificação funcional do sistema. Em outras palavras, o testador está interessado no comportamento do sistema e não na sua implementação. A Figura 4 ilustra a estrutura dos Testes de Caixa-Preta.

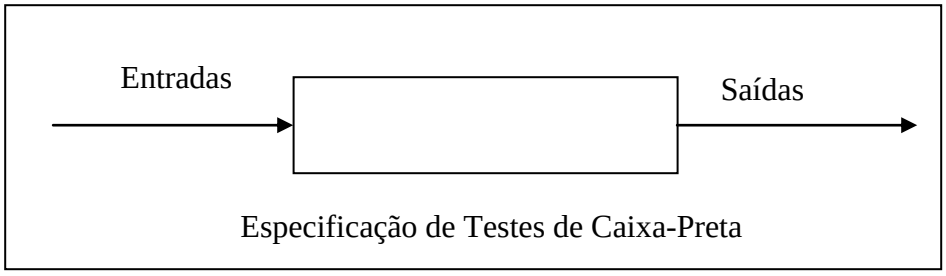

Figura 4 - Testes de Caixa-Preta

Segundo Delamaro (2007) todos os critérios da técnica de teste funcional baseiam-se na especificação do produto testado e a qualidade de tais critérios depende diretamente da existência de uma boa especificação de requisitos. Especificações ausentes ou incompletas podem limitar a aplicação dos critérios funcionais.

Os testes funcionais devem garantir que:

- Todos os tipos ou classes de entradas válidas sejam aceitos e processados corretamente pelo sistema;
- Todos os tipos de entradas inválidas sejam rejeitados e processados corretamente pelo sistema;
- Todas as possíveis classes de saídas do sistema sejam executadas e examinadas;
- Todos estados efetivos e de transição do sistema sejam executados e examinados;
- Todas as funções sejam executadas.

Os testes funcionais são implementados através dos casos de testes, que descrevem a especificação do que será testado, em termos de:

- Condição de execução: é a descrição sucinta do cenário de uso da aplicação que será testado.
- Valores de entrada: é o conjunto de dados de entrada no teste que será capaz de exercitar o sistema conforme a condição descrita.
- Resultado esperado: é o retorno gerado por um determinado comportamento do sistema.

A Figura 5 representa a implementação dos testes funcionais.

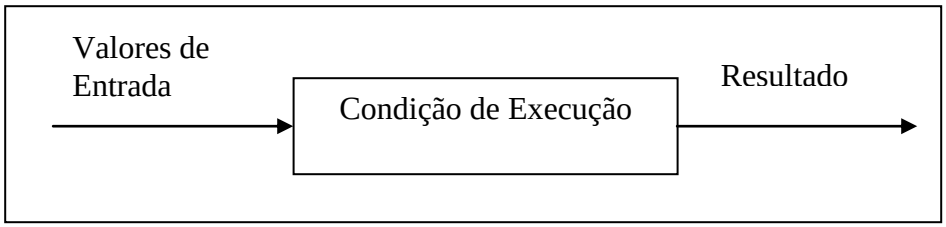

Figura 5 - Implementação dos Testes Funcionais

Uma das formas possíveis de projetar casos de testes é derivando-os a partir da especificação de requisitos, ou casos de uso, isto é, o testador deve conhecer estas especificações e traduzi-las em casos de testes e pontos de verificação (HEUMANN, 2001). Isto possibilita que o planejamento dos testes possa ser realizado ainda durante a fase de especificação dos requisitos do software, independente do desenvolvimento do código-fonte.

3.3 Especificação do Processo de Teste

O processo de teste proposto envolve cinco etapas: planejamento, projeto, execução, análise dos resultados e correção, representadas na Figura 6.

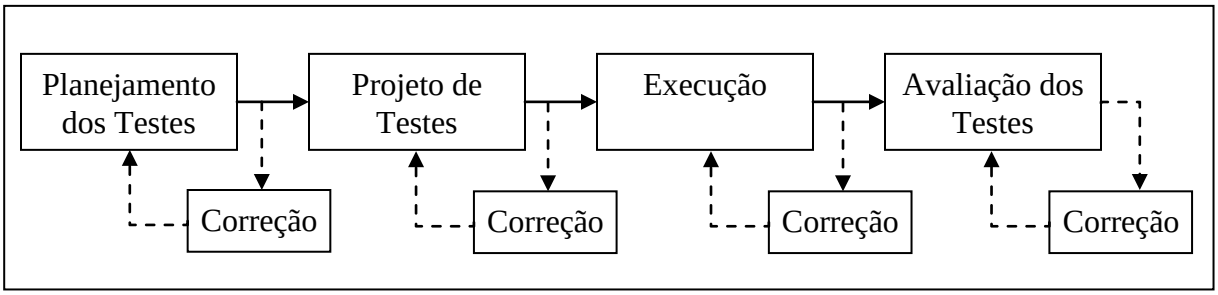

Figura 6 - Etapas do Processo de Teste

Para cada etapa do processo de teste foram definidas as atividades necessárias para que os objetivos do processo possam ser alcançados. Cada etapa possui um propósito e artefatos de entrada e de saída. Os artefatos de entrada podem ser modelos de documentos, documentos preenchidos, padrões do processo ou elementos informativos. Os artefatos de saída podem ser um ou mais artefatos de entrada atualizados.

3.3.1 Planejamento dos Testes

Na etapa de planejamento deve-se definir o escopo dos testes a serem realizados (o que), os responsáveis pela execução do teste (quem), o período previsto para realização (quando) e os recursos necessários para a realização das atividades de testes do software (como). Nesta etapa é elaborado o artefato Plano de Testes de Software, que terá como entrada alguns artefatos de especificação de requisitos, para atender a finalidade do software a ser testado.

A etapa de planejamento do processo do teste deve ser realizada paralelamente a etapa de planejamento do processo de desenvolvimento do software.

Nesta etapa, o Analista de Teste entende os objetivos do software a ser testado e define os critérios para o início do processo de teste, determinando elementos e/ou condições necessárias para iniciar a execução das tarefas.

3.3.1.1 Entradas e Saídas

A Figura 7 apresenta as entradas necessárias para a etapa de planejamento: Documentos de Requisitos, Casos de Uso e Cronograma do Projeto, além do artefato gerado como saída desta fase: o Plano de Testes.

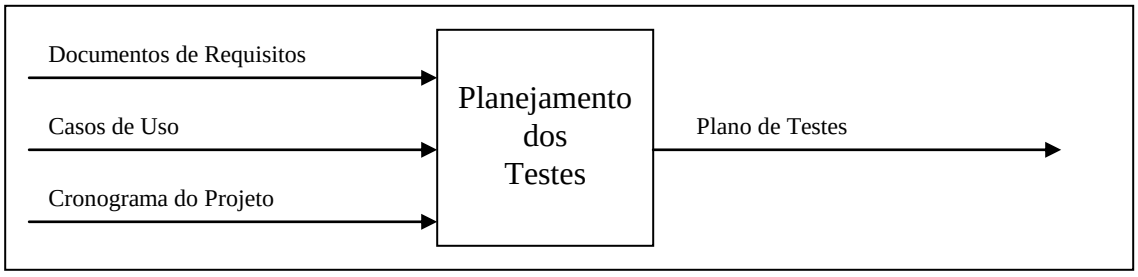

Figura 7 - Entradas e Saídas da Fase de Planejamento

3.3.1.2 Responsáveis

Na etapa de planejamento os responsáveis pela execução das atividades são:

- Gerente de Projeto: responsável por determinar, junto com o Analista de Teste, o escopo dos testes, identificar o que será e o que não será testado e analisar o Plano de Teste;
- Analista de Teste de Software: responsável por definir o escopo dos testes, especificar os artefatos, padrões e ferramentas, identificar os itens que serão e os que não serão testados e elaborar o Plano de Teste.

3.3.1.3 Atividades

As atividades necessárias para produzir as saídas da etapa de planejamento do processo de teste são:

- Definir o escopo dos testes;
- Determinar os artefatos, padrões, ferramentas e ambiente de testes;
- Identificar itens que serão ou não testados;
- Estabelecer abordagens e estratégias de teste que serão aplicadas na execução dos testes, elaborando um fluxograma dessas estratégias;
- Definir ambiente de testes (software a ser testado e as ferramentas de apoio);
- Elaborar o cronograma das atividades;
- Estabelecer responsabilidades dentro da equipe definida;
- Estabelecer treinamentos necessários;
- Identificar riscos;
- Definir o critério de aprovação dos testes;
- Elaborar o Plano de Teste que será aplicado ao software em desenvolvimento.

3.3.2 Projeto de Testes

Os objetivos da etapa de projeto de testes são: identificar, descrever e gerar o modelo de teste e seus artefatos, Procedimentos de Teste e Casos de Teste.

Os Casos de Teste gerados durante a fase de projeto de testes devem possuir dados de entrada e valores de saída esperados, além das pré-condições necessárias para a execução dos Casos de Teste.

O Procedimento de Teste, gerado também nesta fase, contém os passos necessários que devem ser executados para a realização de um conjunto de testes. O procedimento de teste deve conter os passos para a instalação da aplicação a ser testada, instalação de ferramentas de apoio e para a realização de um caso de uso, isto é, o teste funcional.

Para cada requisito de software desenvolvido, devem ser gerados seus respectivos Casos de Teste e Procedimentos de Teste.

3.3.2.1 Entradas e Saídas

A Figura 8 apresenta as entradas necessárias para a etapa de projetos de teste: Documentos de Requisitos, Casos de Uso e Plano de Testes, além dos artefatos gerados como saídas desta fase: Caso de Testes e Procedimento de Testes.

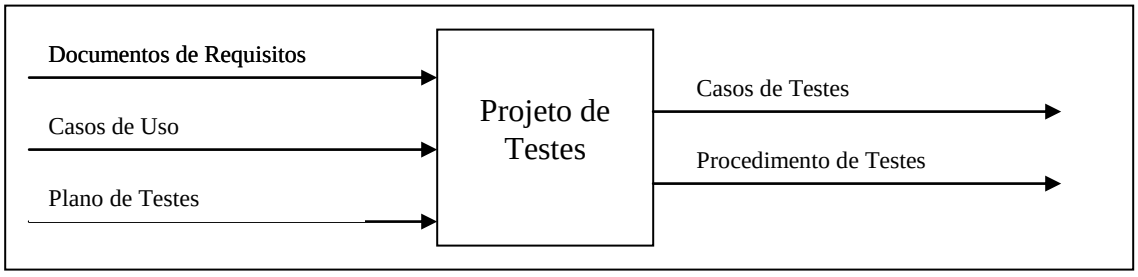

Figura 8 - Entradas e Saídas da Fase de Projeto

3.3.2.2 Responsáveis

Na etapa de projeto de teste os responsáveis pela execução das atividades são:

- Analista de Teste: responsável por elaborar os documentos de Caso de Teste e o Procedimento de Testes. Também é responsável por executar as revisões nos documentos do projeto em desenvolvimento, no caso de serem detectados defeitos durante a fase de projeto de testes, e preparar o ambiente de teste;
- Gerente de Projeto: responsável por analisar e atribuir correções dos documentos do projeto revisados pelo Analista de Teste.

3.3.2.3 Atividades

As atividades necessárias para produzir as saídas da etapa de projeto do processo de teste são:

- Definir a configuração do ambiente de testes;
- Estruturar a implementação dos testes;
- Definir os elementos de hardware e software necessários nas configurações de testes;
- Desenvolver guias de testes;
- Avaliar e verificar os resultados;
- Executar revisões nos documentos relacionados ao desenvolvimento;
- Relatar falhas e inconsistências encontradas nesses documentos;
- Elaborar documentos de Casos de Teste, baseados nos documentos de Casos de Uso;
- Planejar a Execução dos testes, baseada nas estratégias definidas no Plano de Teste;

- Preparar o ambiente de teste, disponibilizando hardware e softwares necessários;
- Apresentar documentos elaborados à equipe e, se necessário, realizar treinamentos.

3.3.3 Execução dos Testes

Na etapa de execução do processo de teste, o objetivo é realizar os testes definidos nas etapas anteriores, garantindo que cada um dos requisitos do software seja testado.

Os testes planejados são executados e os resultados são registrados no documento de Caso de Teste. Os defeitos detectados são descritos no Relatório de Incidente de Testes.

3.3.3.1 Entradas e Saídas

A Figura 9 apresenta as entradas necessárias para a etapa de execução de teste: Casos de Testes e Procedimento de Testes, além dos artefatos gerados como saídas desta fase: Casos de Testes alterados e Relatório de Incidente de Testes. Durante a efetiva execução do software podem ser identificados defeitos nos artefatos Casos de Testes ou testes não planejados, e desta forma durante esta etapa, foi prevista a possibilidade de alteração deste artefato.

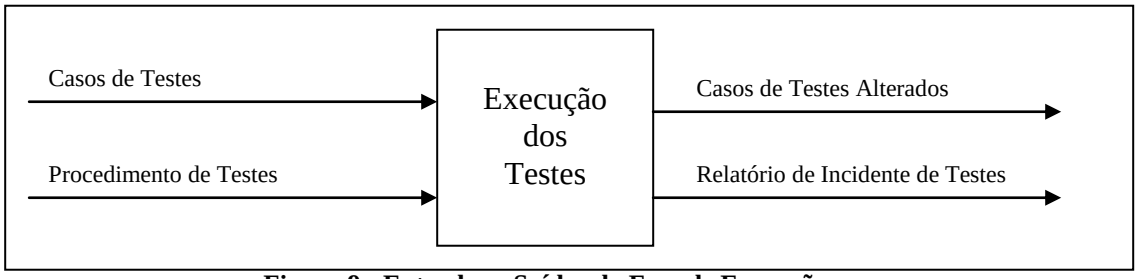

Figura 9 - Entradas e Saídas da Fase de Execução

3.3.3.2 Responsáveis

Na etapa de execução de teste os responsáveis pelas atividades são:

- Analista de Teste: responsável por executar os testes anteriormente apresentados, garantir o uso e preenchimentos dos artefatos de entrada e saída desta etapa e gerar os relatórios de erros;
- Analista de Sistemas: responsável por corrigir os erros detectados.

3.3.3.3 Atividades

As atividades necessárias para produzir as saídas da etapa de execução do processo de teste são:

- Executar os testes planejados nos documentos de Caso de Teste;
- Registrar os testes e relatar os erros encontrados durante as diversas etapas de execução;
- Garantir a qualidade do produto de software desenvolvido.

3.3.4 Avaliação dos Testes

Na etapa de avaliação do processo de teste, é realizado o teste de aceitação do software e são realizadas atividades de avaliação das etapas anteriores.

Nesta etapa as evidências dos resultados obtidos durante as etapas anteriores são analisadas e o software desenvolvido é validado, de forma a identificar se este atende as especificações feitas pelo cliente e se demonstra conformidade com os requisitos.

A avaliação dos resultados permite que os processos de desenvolvimento e de teste de software sejam aprimorados, através do aprendizado obtido com o processo.

3.3.4.1 Entradas e Saídas

A Figura 10 apresenta as entradas necessárias para a etapa de avaliação de testes: Plano de Testes, Casos de Testes, Procedimento de Testes e Relatório de Incidente de

Testes, além dos artefatos gerados como saídas desta fase: Relatório de Resumo de Testes e Plano de Melhorias.

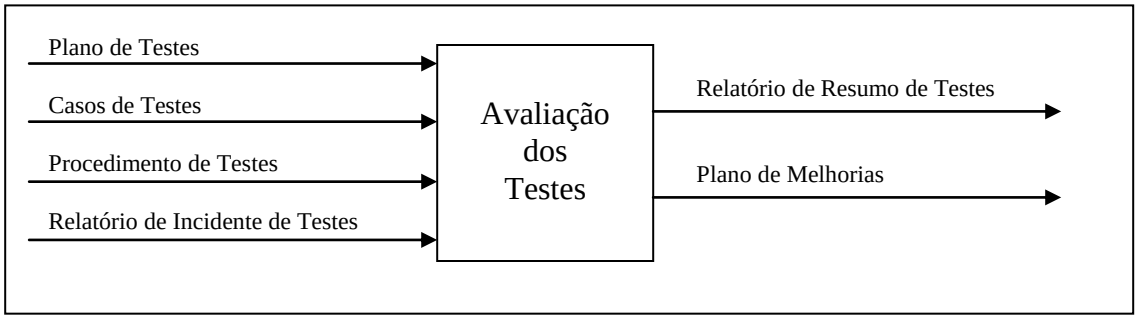

Figura 10 - Entradas e Saídas da Fase de Avaliação

3.3.4.2 Responsáveis

Na etapa de avaliação de teste os responsáveis pelas atividades são:

- Analista de Teste: responsável por analisar os erros detectados durante o teste de aceitação realizado pelo cliente/usuário, e encaminhar os erros detectados aos responsáveis pela correção. Deverá criar o artefato Relatório de Resultado dos Testes, a partir da análise das demais etapas do processo de teste. Ainda é de sua responsabilidade avaliar os resultados dos testes e propor melhorias para futuros projetos;

- Gerente de Projeto: responsável por avaliar, junto com o Analista de Teste, o resultado obtido nas demais etapas do processo de teste e criar o documento Plano de Melhorias;

- Analista de Sistemas: responsável por corrigir os erros detectados durante o teste de aceitação do sistema.

3.3.4.3 Atividades

As atividades necessárias para produzir as saídas da etapa de avaliação do processo de teste são:

- Realizar testes de aceitação com usuários responsáveis para verificar se o produto desenvolvido atende aos requisitos pré-definidos;

- Corrigir erros encontrados durante a realização dos testes de aceitação, realizados pelo cliente/usuário;
- Avaliar os resultados dos testes registrados nos documentos de Caso de Teste e Relatório de Incidente de Testes.
- Gerar Relatório de Resultado dos Testes;
- Criar documento Plano de Melhorias com sugestões de mudanças no processo de desenvolvimento e de teste, baseados nos resultados obtidos.

3.3.5 Correção

Na etapa de correção do processo de teste os artefatos desenvolvidos durante as demais etapas do processo (planejamento, projeto, execução e avaliação) são corrigidos, de acordo com os defeitos identificados nas inspeções realizadas nestes artefatos.

A etapa de correção também envolve a solução dos erros encontrados durante a execução do software na etapa de execução de testes.

3.3.5.1 Entradas e Saídas

A Figura 11 apresenta as entradas necessárias para a etapa de correção de testes: Plano de Testes, Casos de Testes, Procedimento de Testes, Relatório de Incidente de Testes, Relatório de Resumo de Testes e Registro de Inspeção. Todos os documentos que foram gerados como saídas das demais etapas do processo de teste servirão com entrada para a fase de correção, dependendo do momento a ser executado, e o mesmo artefato, após a sua correção, poderá ser considerado uma saída da etapa de correção. O código do programa corrigido também é um dos artefatos de saída gerados nesta etapa.

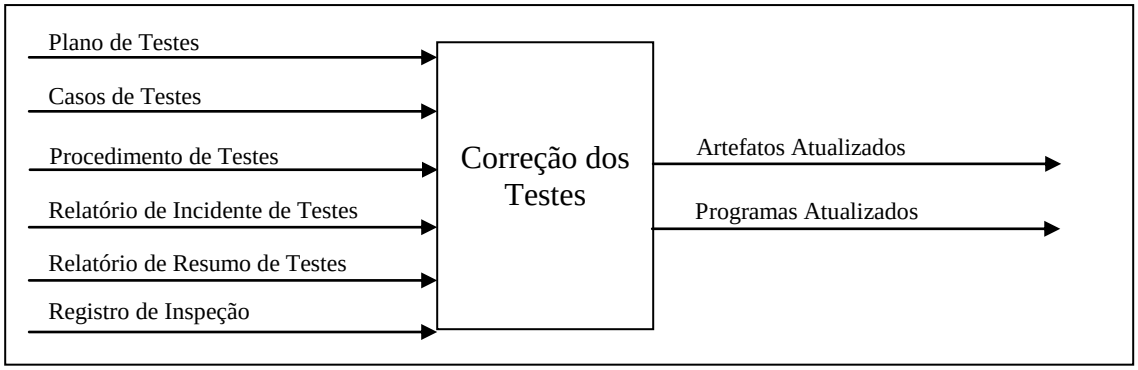

Figura 11 - Entradas e Saídas da Fase de Correção

3.3.5.2 Responsáveis

Na etapa de correção de teste os responsáveis pelas atividades são:

- Analista de Teste: responsável por corrigir os defeitos detectados durante a inspeção dos artefatos de teste e por atualizar o documento Plano de Melhorias com sugestões de mudanças nos artefatos criados durante o processo de teste.

- Analista de Sistema: responsável por corrigir os erros de codificação identificados na etapa de execução de testes de software, registrados no artefato Relatório de Incidente de Testes.

3.3.5.3 Atividades

As atividades necessárias para produzir as saídas da etapa de correção do processo de teste são:

- Análise dos defeitos encontrados durante a execução da inspeção nos artefatos de testes desenvolvidos e registrados no artefato Registro de Inspeção;

- Corrigir os defeitos listados no artefato Registro de Inspeção;

- Criar documento Plano de Melhorias com sugestões de mudanças na criação de novos artefatos de teste, baseados nos resultados obtidos.

3.4 Considerações Finais

Neste capítulo foram apresentados os objetivos do processo de teste proposto, além de sua especificação detalhada. Foram descritas as atividades a serem executadas e suas respectivas responsabilidades, procedimentos e os artefatos de testes gerados durante a execução do processo de teste. O próximo capítulo apresenta a especificação detalhada do processo de inspeção a ser executado nos artefatos gerados durante o teste funcional de software.

4 PROCESSO DE INSPEÇÃO EM ARTEFATOS DE TESTES FUNCIONAIS

Neste capítulo são apresentados os objetivos da implantação do processo de inspeção em artefatos gerados durante a execução dos testes funcionais de software, além da descrição das atividades executadas durante o processo, os responsáveis, e a especificação das entradas e saídas geradas.

4.1 Objetivos do Processo de Inspeção

Os objetivos do processo de inspeção em artefatos de testes funcionais são:

- Verificar se os artefatos gerados durante a fase de teste de software estão seguindo o padrão especificado;
- Verificar se o artefato produzido atende as especificações dos produtos de trabalho antecessores;
- Fornecer meios de mensurar o nível de assimilação a qualquer assunto relacionado ao esforço de melhoria de processo;
- Sugerir oportunidades de melhoria para o autor do artefato que está sendo inspecionado.

Os defeitos encontrados durante o processo de inspeção não serão corrigidos durante a sua execução, e sim encaminhados para que o responsável pelo artefato corrija-os posteriormente.

4.2 Especificação do Processo de Inspeção

O processo de inspeção em artefatos de testes funcionais é composto pelas seguintes etapas: Planejamento, Preparação, Reunião de Inspeção, Correção e Finalização, conforme apresentado na Figura 12.

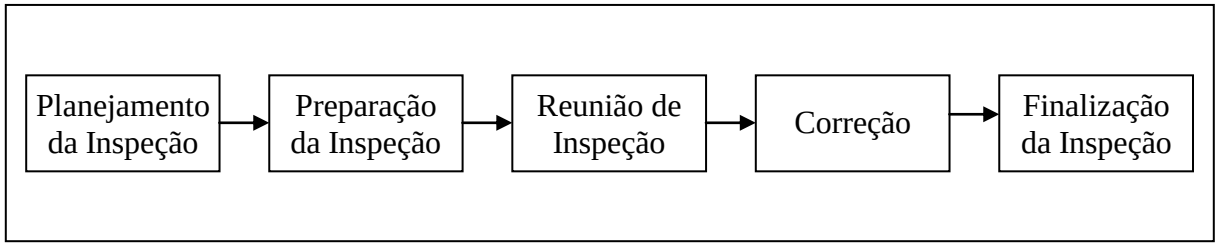

Figura 12 - Processo de Inspeção de Testes Funcionais

4.2.1 Papéis e Responsabilidades

Durante o processo de inspeção, cada um dos participantes envolvidos pode receber um ou mais papéis e responsabilidades. Os papéis de uma inspeção são distribuídos em: Autor, Moderador, Leitor, Escritor, Inspetor e Coordenador das Inspeções.

Segundo o padrão IEEE 1012 (IEEE, 1998b) o processo de inspeção deve ser composto por um grupo de três a seis pessoas, sendo um deles o Autor. Os demais serão responsáveis efetivamente pela inspeção.

O Autor é a pessoa responsável pelo desenvolvimento do artefato a ser inspecionado. É sua responsabilidade entregar os produtos de trabalho e documentos associados aos participantes da inspeção, além de esclarecer as dúvidas relativas ao produto a ser inspecionado.

O papel do Moderador é ser um facilitador durante o processo de inspeção. Junto com o Autor, ele irá identificar os demais participantes da inspeção e planejar o cronograma de inspeção. Durante a reunião o Moderador utilizará uma lista de verificação para auxiliar a inspeção.

O Leitor é responsável por fazer a leitura dos artefatos inspecionados, permitindo que o time de inspeção apresente comentários, não-conformidades ou questionamentos.

O Escritor é responsável por registrar e classificar as não-conformidades encontradas durante a inspeção.

O papel do Inspetor é identificar defeitos, desvios e sugerir melhorias no artefato que está sendo inspecionado.

O Coordenador das Inspeções é o responsável pelas métricas de inspeção do projeto, mantendo os registros das inspeções conduzidas e dados do resumo de cada inspeção, gerando os relatórios de inspeção.

No estudo realizado por Denger e Shull (2007) foi proposto que todos os envolvidos no processo de inspeção incorporassem a visão do usuário final também durante as atividades de inspeção, permitindo aos inspetores identificar de modo mais eficaz defeitos existentes.

4.2.2 Planejamento

Na fase de Planejamento do processo de inspeção, o Autor entrega ao Moderador o artefato produzido e os documentos de suporte utilizados para a criação do artefato.

O cronograma da inspeção será montado considerando o tamanho e complexidade dos produtos de trabalho entregues.

O Autor e o Moderador são responsáveis por selecionar a equipe de inspeção e atribuir seus papéis, além de agendar a reunião de inspeção e estimar o tempo de preparação.

Na atividade final da fase de Planejamento, o Autor deve distribuir os artefatos necessários para inspeção, para toda a equipe de inspeção.

A Figura 13 ilustra os responsáveis, as entradas e saídas geradas durante a fase de Planejamento da Inspeção.

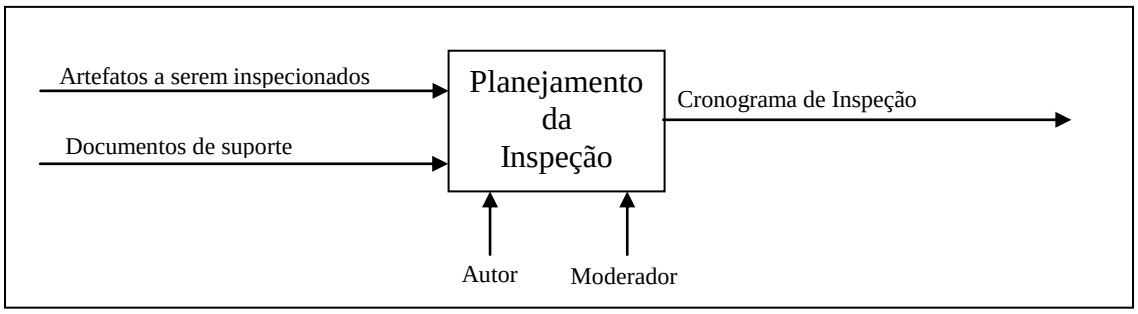

Figura 13 - Fase de Planejamento da Inspeção

4.2.3 Preparação

Na fase de Preparação os Inspetores analisam o produto de trabalho em busca de não-conformidades e fazem as anotações necessárias para serem expostas durante a Reunião de Inspeção, conforme apresentado na Figura 14.

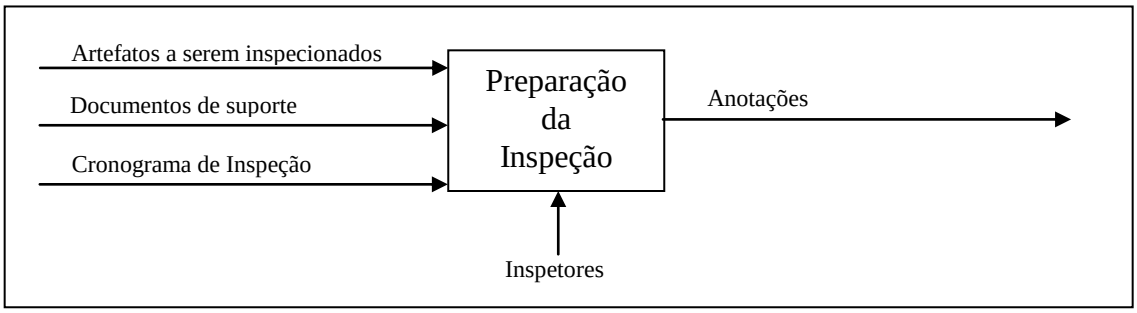

Figura 14 - Fase de Preparação da Inspeção

4.2.4 Reunião de Inspeção

Na Reunião de Inspeção o Leitor faz a leitura do artefato inspecionado.

Para facilitar o levantamento de defeitos, a leitura do artefato pode ser feita em pequenas partes e, em seguida, os inspetores discutem sobre os defeitos e não-conformidades encontradas.

O papel do Escritor é registrar as não-conformidades no formulário Registro de Inspeção, identificando-as por tipo (Faltando, Errada, Extra, Usabilidade, Performance, Não-defeito), Severidade (Principal ou Secundária), Localização e Descrição.

Durante a reunião, o Autor deverá responder aos questionamentos sobre o produto de trabalho em inspeção, e o Moderador irá validar o artefato, seguindo a respectiva lista de verificação.

Ao final da Reunião de Inspeção a equipe de inspeção deve decidir o *status* do artefato inspecionado, podendo este ser Aceito, Condicionalmente aceito, Re-Inspeção, ou Inspeção não-completada.

A Figura 15 ilustra as entradas e saídas geradas na fase de Reunião de Inspeção, além dos responsáveis envolvidos.

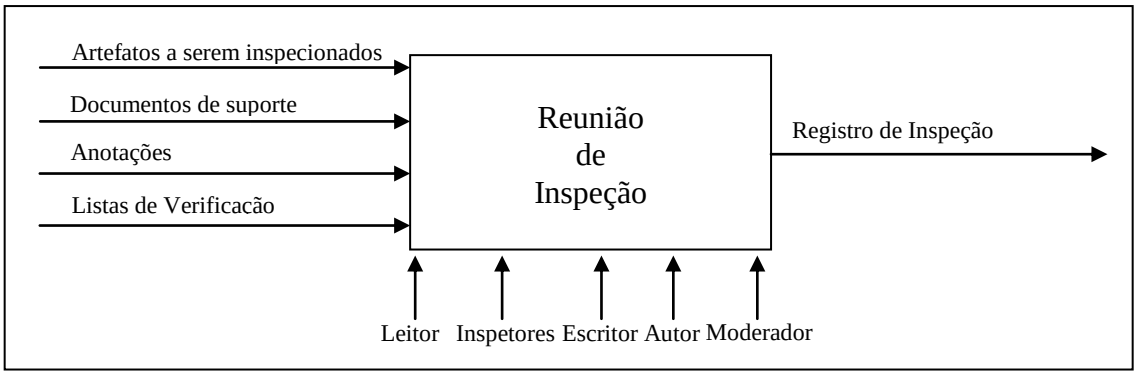

Figura 15 - Fase de Reunião de Inspeção

4.2.4.1 Seleção da Técnica de Inspeção

No capítulo 2 foram apresentadas diversas técnicas de inspeção de software e a característica que diferencia essas técnicas é o nível de formalidade usado durante a atividade de identificação de defeitos. O nível de formalidade permite categorizar as inspeções como: *Ad-hoc*, Lista de Verificação ou *Checklist* e Técnicas de Leitura.

A técnica *Ad-hoc* é considerada a mais simples para identificar defeitos em artefatos de software. Ela não oferece nenhum tipo de apoio ou procedimento de execução formal e sistemático de inspeção. Sendo assim, o conhecimento e a experiência do inspetor influenciam diretamente no número de defeitos identificados, apresentando variações nos defeitos encontrados quando a inspeção é executada por diferentes pessoas (BERLING, 2004).

A técnica de Lista de Verificação ou *Checklist* oferece maior suporte do que a técnica *Ad-hoc*, pois fornece ao inspetor um conjunto de questionamentos que o auxiliam a avaliar detalhadamente o artefato na busca por defeitos (BERLING, 2004).

Utilizar esse tipo de técnica em inspeções de artefatos gerados durante o teste de software consiste em definir questionamentos que, por exemplo, indicariam ao inspetor como identificar os elementos que devem ser analisados, visando avaliar a qualidade destes artefatos.

As Técnicas de Leitura são outro tipo de inspeção e consistem em procedimentos que visam guiar individualmente os inspetores no entendimento de um artefato de software e, por conseqüência, na identificação de defeitos (SHULL, 1998). Estudos indicam que as Técnicas de Leitura se mostram mais eficientes na detecção de defeitos quando comparadas a outras técnicas de inspeção, como as Listas de Verificação ou *Ad-hoc* (SHULL, 1998; BERLING, 2004; THELIN, 2004).

As Técnicas de Leitura desenvolvidas focam a inspeção em artefatos gerados durante a fase de requisitos dentro de um processo de desenvolvimento de software (BASILI, 1996; SHULL, 1998; CIOLKOWSKI, 2002; THELIN, 2004) e diagramas com notação UML (TRAVASSOS, 1999). Não foram identificadas, na revisão bibliográfica, técnicas de leitura desenvolvidas especificamente para artefatos gerados durante a fase de testes de software.

Para que esse tipo de técnica seja utilizado, o artefato deve ser representado em uma forma específica e padronizado, de acordo com a Técnica de Leitura a ser aplicada, e exige certo grau de experiência por parte dos inspetores.

Sendo assim, para o desenvolvimento deste trabalho, foi utilizada a técnica de inspeção baseada em Listas de Verificação, pois visa atender inspetores que ainda não possuem alto grau de experiência na função. Os critérios para a criação das Listas de Verificação utilizadas no processo proposto serão abordados mais adiante, ainda neste capítulo.

4.2.5 Correção

Na fase de Correção o Autor corrige as não-conformidades encontradas, registrando-as no Registro de Inspeção, a ação tomada.

O Autor pode corrigir qualquer outro problema baseado nos defeitos encontrados durante a inspeção.

A Figura 16 ilustra as entradas e saídas geradas durante a fase de Correção da Inspeção, além do responsável envolvido.

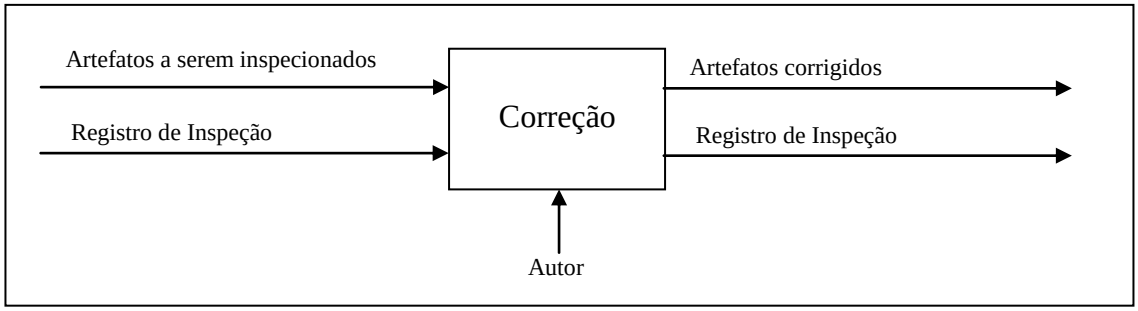

Figura 16 - Fase de Correção da Inspeção

4.2.6 Finalização

Na fase de Finalização o Moderador deverá verificar se todas as não-conformidades foram corrigidas, e deve encaminhar o Relatório de Inspeção ao Coordenador de Inspeções. Após a finalização o Autor poderá disponibilizar os artefatos produzidos para as fases seguintes do projeto, conforme apresentado na Figura 17.

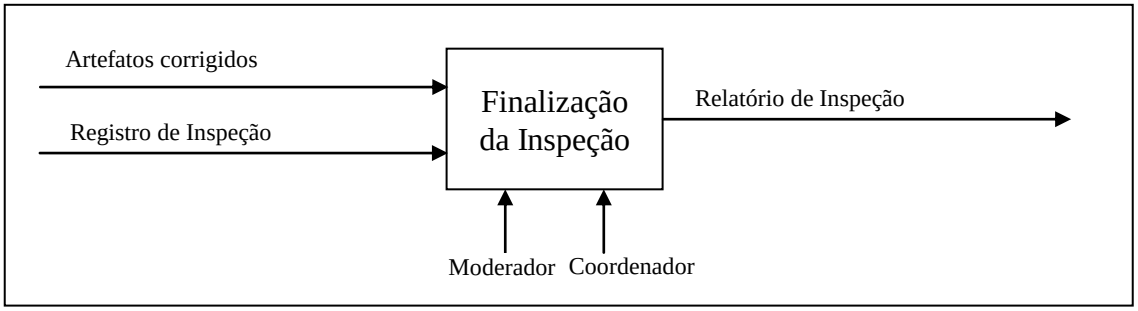

Figura 17 - Fase de Finalização da Inspeção

4.3 Listas de Verificação

Segundo Gilb e Graham (1993 apud GREGOLIN, 2007) para a criação de listas de verificação devem ser utilizadas as regras de elaboração do produto a ser inspecionado, associando a cada questão da lista de verificação a regra da qual é derivada. Para os autores, as listas de verificação orientam os inspetores sobre o quê inspecionar e detalham as regras de elaboração do artefato.

Gregolin (2007) sugere que as questões de uma lista de verificação devem se concentrar nos maiores defeitos, não havendo a necessidade de conter todas as questões possíveis, e que as questões devem ser diretas e objetivas e formatadas de modo que a resposta negativa represente a identificação de um problema.

Os requisitos que devem ser atendidos para a criação das listas de verificação estão apresentados na Tabela 7 e foram adaptados a partir dos requisitos de qualidade de documentação do usuário da norma ISO/IEC 12119, descritos no capítulo 2 desta dissertação.

Item	Requisitos
Completude	A documentação de testes deve possuir todas as informações necessárias para a execução dos testes, tais como planejamento das atividades de teste, recursos necessários.
Correção	As informações apresentadas nos artefatos de teste devem estar corretas e sem ambigüidade.
Consistência	Os testes previstos devem contemplar todos os requisitos definidos para o software, identificando entradas e saídas válidas e inválidas, devendo existir coerências entre os artefatos gerados durante a fase testes e de requisitos de software. Os artefatos gerados durante a fase de testes deverão permitir rastreabilidade com outros documentos de projeto.
Inteligibilidade	A documentação deve ser compreensível por todos os envolvidos durante o processo de desenvolvimento do software (gerente do projeto, analistas de sistemas, analistas de teste, usuários finais e outros), utilizando termos apropriados e explicações detalhadas.
Apresentação e organização	Deve ser apresentada através de uma forma que facilite uma visão geral, através de índices e tabelas de conteúdo.

Tabela 7 - Requisitos de Qualidade para criação das Listas de Verificação

O padrão IEEE 829 (IEEE, 1998a) descreve as regras para a elaboração dos artefatos gerados durante o processo de teste de software e serviu como base para a criação das questões das Listas de Verificação utilizadas na inspeção dos artefatos propostos. O padrão IEEE 1028 (IEEE, 1997) que descreve as regras que devem ser seguidas para a execução do

processo de inspeção também foi avaliado para a criação da Lista de Verificação. As questões propostas para a inspeção dos artefatos e o relacionamento delas com os Requisitos de Qualidade serão detalhadas adiante.

4.3.1 Lista de Verificação do Artefato Plano de Testes

A Tabela 8 apresenta a lista de questões propostas para a inspeção do artefato Plano de Testes e o relacionamento das questões com os Requisitos de Qualidade.

	Questão	Requisito de Qualidade
1	Foi criado um identificador único para o plano de testes.	Consistência, Apresentação e organização
2	O escopo do plano de testes está claramente definido e delimitado.	Completude, Consistência
3	Os documentos de projeto utilizados para gerar o plano foram corretamente identificados.	Consistência
4	Foram identificados os sistemas, áreas e interfaces envolvidas.	Completude
5	As funcionalidades a serem testadas foram identificadas de forma clara e precisa.	Completude, Consistência
6	As funcionalidades a não serem testadas foram identificadas de forma clara e precisa e foram incluídas as justificativas.	Completude, Consistência
7	Os itens a serem testados foram listados de forma clara e precisa.	Completude, Consistência
8	Os requisitos a serem testados estão devidamente identificados de acordo com o Documento de Especificação de Requisitos.	Completude, Consistência
9	Estão identificados os critérios de aceite/falha para todos os requisitos identificados para teste, de forma clara e precisa e são suficientes e aceitáveis.	Completude, Consistência, Inteligibilidade
10	A estratégia de teste documenta os tipos de testes a serem implementados e executados.	Completude, Consistência
11	Os tipos de testes a serem implementados possuem os objetivos, técnicas e critérios de finalização.	Completude, Consistência, Inteligibilidade
12	O plano de teste identifica os artefatos criados pelas atividades de testes e datas de inicio e fim para criação dos artefatos.	Completude, Consistência
13	As atividades de testes foram relacionadas, identificando datas de início e fim e esforço necessários.	Completude, Consistência
14	A lista de atividades é consistente com o processo adotado para os testes.	Consistência
15	A lista de atividades é consistente com os aspectos a testar.	Consistência
16	Foram identificados os recursos necessários para implementar e executar os testes, incluindo hardware, software e recursos humanos.	Completude, Consistência, Inteligibilidade
17	O plano de testes identifica os riscos ou contingências que podem afetar ou impactar o esforço de teste.	Completude, Consistência, Inteligibilidade
18	O plano de testes está completo, correto e não ambíguo.	Correção

Tabela 8 - Listas de Verificação do Plano de Teste e os Requisitos de Qualidade

4.3.2 Lista de Verificação do Artefato Casos de Testes

A Tabela 9 apresenta a lista de questões propostas para a inspeção do artefato Casos de Testes e o relacionamento das questões com os Requisitos de Qualidade.

	Questão	Requisito de Qualidade
1	Foi criado um identificador único para o Caso de Testes.	Consistência, Apresentação e organização
2	O escopo do caso de testes está claramente definido e delimitado.	Completude, Consistência
3	Os documentos utilizados para gerar os casos de teste foram corretamente identificados.	Consistência
4	O documento de casos de testes contém o nome do Caso de Testes, relacionado com o Caso de Uso correspondente, permitindo a rastreabilidade entre eles.	Completude, Consistência
5	Para cada requisito de teste estão descritos os cenários de testes, fluxo básico e fluxos alternativos	Completude, Consistência, Correção
6	O caso de testes contém os procedimentos de teste que serão usados para executar o caso de testes	Completude, Consistência
7	Os casos de teste identificados são suficientes para atender as funcionalidades a serem testadas, previstas no plano de testes.	Completude, Consistência

Tabela 9 - Listas de Verificação dos Casos de Teste e os Requisitos de Qualidade

4.3.3 Lista de Verificação do Artefato Procedimentos de Testes

A Tabela 10 apresenta a lista de questões propostas para a inspeção do artefato Procedimentos de Testes e o relacionamento das questões com os Requisitos de Qualidade.

	Questão	Requisito de Qualidade
1	Foi criado um identificador único para o Procedimento de Testes.	Consistência, Apresentação e organização
2	O escopo do procedimento de testes está claramente definido e delimitado.	Completude, Consistência
3	O documento de procedimentos de testes contém o nome do Procedimento de Testes, relacionado com o Caso de Teste correspondente	Consistência
4	Estão relacionados os documentos de referência (casos de usos, especificação de requisitos, casos de teste, procedimentos de testes)	Completude, Consistência
5	Estão definidos os requisitos especiais para a execução do Procedimento de Testes	Completude, Consistência, Correção
6	Foi descrito um fluxo passo a passo, com as atividades a serem executadas para a execução do Procedimento de Testes	Completude, Consistência
7	O fluxo é detalhado o suficiente permitindo ser executado manualmente, ou convertido em um script de teste.	Completude, Consistência, Correção
8	Estão descritas as entradas esperadas	Completude, Consistência, Correção
9	Estão descritas as saídas esperadas	Completude, Consistência, Correção
10	As entradas e saídas esperadas são suficientes e realistas	Completude, Consistência, Correção
11	Foram definidos os métodos para verificação dos valores esperados de forma clara e precisa	Completude, Consistência, Correção

Tabela 10 - Listas de Verificação dos Procedimentos de Teste e os Requisitos de Qualidade

4.3.4 Lista de Verificação do Artefato Relatório de Incidente de Testes

A Tabela 11 apresenta a lista de questões propostas para a inspeção do artefato Relatório de Incidente de Testes e o relacionamento das questões com os Requisitos de Qualidade.

	Questão	Requisito de Qualidade
1	Foi criado um identificador único para o relatório de Incidente de Testes.	Consistência, Apresentação e organização
2	Estão relacionados os documentos de referência (casos de usos, especificação de requisitos, casos de teste, procedimentos de testes, incidentes de testes).	Completude, Consistência
3	Estão identificadas a data e hora de realização dos testes.	Consistência
4	Estão identificados os nomes do testador e demais pessoas envolvidas no teste.	Completude, Consistência
5	Estão descritos os ambientes de hardware e software utilizados para a execução dos testes.	Completude, Consistência, Correção
6	Foram identificados os Casos de Testes executados que geraram o documento de Relatório de Incidentes.	Completude, Consistência

Tabela 11 - Listas de Verificação do Relatório de Incidente de Teste e os Requisitos de Qualidade

4.3.5 Lista de Verificação do Artefato Relatório de Resumo de Testes

A Tabela 12 apresenta a lista de questões propostas para a inspeção do artefato Relatório de Resumo de Testes e o relacionamento das questões com os Requisitos de Qualidade.

	Questão	Requisito de Qualidade
1	Foi criado um identificador único para o relatório de Incidente de Testes.	Consistência, Apresentação e organização
2	Estão relacionados os documentos de referência (casos de usos, especificação de requisitos, casos de teste, procedimentos de testes, incidentes de testes).	Completude, Consistência
3	Estão identificadas a data e hora de realização dos testes.	Consistência
4	Estão identificados os nomes do testador e demais pessoas envolvidas no teste.	Completude, Consistência
5	Estão descritos os ambientes de hardware e software utilizados para a execução dos testes.	Completude, Consistência, Correção
6	Foram identificados os Casos de Testes executados que geraram o documento de Relatório de Incidentes.	Completude, Consistência

Tabela 12 - Listas de Verificação do Relatório de Resumo de Teste e os Requisitos de Qualidade

4.3.6 Avaliação dos Artefatos

Durante a execução do processo de inspeção são aplicadas as Listas de Verificação propostas. A aplicação das Listas de Verificação é composta por duas atividades: Diagnóstico e Análise do Resultado.

Na atividade de Diagnóstico as respostas constantes nas Listas de Verificação devem ser respondidas com as opções “Sim” para o item verificado no artefato e aceito, “Não” para o item verificado no artefato e que apresente informação incorreta, ambígua ou incompleta, ou “Não se Aplica” quando o item verificado não for aplicável ao artefato avaliado.

Na atividade de Análise do Resultado, após o preenchimento dos itens de verificação, é efetuado um cálculo com o objetivo de gerar uma nota para a avaliação do artefato inspecionado. Para respostas obtidas com “Sim” considera-se 1 ponto. As respostas “Não” geram 0 ponto, e para as respostas com resultado “Não se aplica” a pergunta não é considerada para efeito de cálculo. Após o preenchimento dos itens de verificação, os resultados obtidos são contabilizados e será aplicado o cálculo descrito na Figura 18 para obter a nota final de avaliação do artefato inspecionado.

$$\text{Nota de Avaliação do Artefato} = \frac{\text{Total de Respostas “Sim”} * 10}{(\text{Total de Respostas “Sim”} + \text{Total de Respostas “Não”})}$$

Figura 18 – Cálculo da Nota de Avaliação do Artefato

O critério para classificação das notas obtidas está demonstrado na Tabela 13 e servirá como instrumento de avaliação durante a Reunião de Inspeção, para que os participantes definam o *status* do artefato inspecionado, podendo ser Aceito, Condicionalmente Aceito, Re-Inspeção, Não Aceito ou Inspeção não-completada.

Nota Obtida	Classificação do Artefato	*Status* Final do Artefato
De 0 a 2	Não Aplicável	Não Aceito/ Re-Inspeção/Não Completada
De 2,1 a 4	Ruim	Não Aceito/ Re-Inspeção/ Não Completada
De 4,1 a 6	Regular	Condicionalmente Aceito/ Re-Inspeção
De 6,1 a 8	Bom	Aceito
De 8,1 a 10	Ótimo	Aceito

Tabela 13 – Critérios de Classificação dos Artefatos Inspecionados

O *status* dos artefatos classificados "Bom" ou "Ótimo" pode ser definido como Aceito, enquanto os artefatos classificados em "Regular" podem ser definidos como Condicionalmente Aceito ou Re-Inspeção e os artefatos classificados como "Ruim" ou "Não Aplicável" podem ter o *status* definido como Re-Inspeção, Não Aceito ou Inspeção não-completada, dependendo de cada caso.

4.4 Métricas do Processo de Inspeção

As métricas coletadas durante o processo de inspeção ajudam a avaliar a eficácia desta atividade quanto à detecção de defeitos, além de verificar se os projetos estão dedicando um montante de esforço adequado às atividades de revisão. As métricas também permitem avaliar o custo das inspeções, através do controle do esforço dedicado para a sua execução e identificar problemas no próprio processo de inspeção, fornecendo sugestões de melhorias (AURUM, 2002).

Exemplos de métricas coletadas durante o processo de inspeção:

- Densidade de defeito = Total de Defeitos / Tamanho atual
- Total de Defeitos = Defeitos Principais + Defeitos secundários
- Esforço de Inspeção = Esforço Planejamento + Esforço Visão Geral + Esforço Preparação + Esforço Reunião + Esforço Re-trabalho
- Taxa de Preparação = Tamanho Planejado / (Esforço Preparação/ N de participantes)
- Taxa de Inspeção = Tamanho atual / Tempo de Reunião de Inspeção

4.5 Considerações Finais

Neste capítulo foram apresentados os objetivos da implantação do processo de inspeção em artefatos gerados durante a execução dos testes funcionais de software e a especificação detalhada do processo. Foram identificadas as atividades executadas durante o processo, os responsáveis e a especificação das entradas e saídas geradas. Foram descritas as Listas de Verificação propostas e os respectivos requisitos de qualidades associados, além descrição dos critérios para avaliação dos artefatos inspecionados.

O próximo capítulo apresenta um comparativo das listas de verificação propostas e listas de verificação identificadas na bibliografia consultada, além do estudo de caso da aplicação do processo de teste proposto e da inspeção nos artefatos de testes gerados e os resultados obtidos com os processos.

5 AVALIAÇÃO DO PROCESSO DE INSPEÇÃO

Neste capítulo é apresentado um comparativo, entre as listas de verificação do processo de inspeção proposto neste trabalho e as listas de verificação identificadas em processos de desenvolvimento de software existentes na indústria de software. É apresentado também um estudo de caso realizado para avaliar as aplicações dos processos de teste e de inspeção em artefatos de testes, além dos resultados obtidos com o processo.

5.1 Comparativo das Listas de Verificação entre o Processo Proposto, RUP e OpenUP

Para avaliar as listas de verificação propostas neste trabalho foi realizado um comparativo com as listas de verificação existentes nos processos de desenvolvimento de software RUP (*Rational Unified Process)*, criado pela *Rational Software Corporation*, e o OpenUP (*Open Unified Process*), mantido pelo Projeto Eclipse e voltado a equipes menores, ambos baseados no UP (*Unified Process*) (OPENUP, 2008; RUP, 2005).

A Tabela 14 apresenta os tipos de artefatos inspecionados, que possuem listas de verificação descritas no processo de inspeção proposto, em comparação com o RUP e OpenUP. A identificação "Sim" corresponde à existência da lista de verificação do artefato no processo comparado, e o termo "Não" indica a inexistência da lista de verificação para o artefato.

Artefato Inspecionado	Processo Proposto	RUP	OpenUP
Plano de Testes	Sim	Sim	Não
Casos de Testes	Sim	Sim	Sim
Procedimentos de Testes	Sim	Sim	Sim
Relatório de Incidente de Testes	Sim	Não	Não
Resumo de Testes	Sim	Não	Não

Tabela 14 – Listas de Verificação Existentes por Artefato Inspecionado

A Tabela 15 apresenta os itens analisados na aplicação da lista de verificação do artefato Plano de Teste, comparando a lista de verificação proposta neste trabalho com a lista de verificação utilizada pelo RUP. O OpenUP não apresenta uma lista de verificação para este artefato e por isso não foi considerado na comparação.

As questões presentes nas listas de verificação foram agrupadas de acordo com o assunto relacionado e formaram a coluna identificada na tabela como "Item Analisado". Para o preenchimento da tabela foi avaliada a existência ou não de alguma questão na lista de verificação referente ao assunto analisado, e embora muitas vezes o artefato possua no seu conteúdo o item analisado, não foi identificada na lista de verificação uma questão específica ao assunto.

Item Analisado	Lista de Verificação Proposta	RUP
Identificação do artefato	X	
Definição do escopo	X	X
Identificação dos artefatos de entrada	X	X
Identificação dos sistemas, áreas e interfaces envolvidas	X	
Identificação das funcionalidades/itens testados	X	X
Identificação das funcionalidades não testadas	X	
Identificação dos requisitos de softwares relacionados ao teste	X	X
Definição dos critérios de aceite ou falha	X	
Identificação dos objetivos, técnicas e critérios de finalização para cada tipo de teste definido	X	
Definição das listas de atividades de testes, artefatos gerados e o cronograma previsto	X	X
Identificação dos recursos necessários para a execução dos testes	X	X
Definição dos riscos relacionados às atividades de teste	X	X
Completude, correção e ambigüidade	X	

Tabela 15 – Itens Analisados na Lista de Verificação do Plano de Teste do Processo Proposto e RUP

A Tabela 16 demonstra o comparativo referente ao artefato Casos de Testes e a Tabela 17 é referente ao artefato Procedimentos de Testes. Não foram criados comparativos para as listas de verificação geradas para os artefatos Relatório de Incidente de Testes e Resumo de Testes, pois não foram identificados no RUP e OpenUP listas de verificações correspondentes a estes artefatos.

Item Analisado	Lista de Verificação Proposta	RUP	OpenUP
Identificação do artefato	X	X	X
Definição do escopo	X		
Identificação dos artefatos de entrada	X	X	X
Identificação da rastreabilidade entre Caso de Teste e o Caso de Uso	X	X	X
Definição dos cenários de testes, fluxo básico e fluxos alternativos necessários	X	X	
Identificação dos Procedimentos de Testes necessários	X	X	X
Completude	X		

Tabela 16 – Itens Analisados na Lista de Verificação dos Casos de Teste do Processo Proposto, RUP e OpenUP

Item Analisado	Lista de Verificação Proposta	RUP	OpenUP
Identificação do artefato	X	X	X
Definição do escopo	X		
Identificação dos artefatos de entrada	X		
Identificação da rastreabilidade entre Procedimento de Teste e o Caso de Teste	X	X	X
Identificação dos requisitos especiais	X		
Detalhamento do fluxo de execução	X		
Identificação das entradas esperadas	X		
Identificação das saídas esperadas	X		
Definição dos métodos de verificação dos resultados	X		

Tabela 17 – Itens Analisados na Lista de Verificação dos Procedimentos de Teste do Processo Proposto, RUP e OpenUP

5.2 Contexto do Estudo de Caso

Para a avaliação dos processos propostos, foi realizado um estudo de caso em um dos sistemas desenvolvidos por uma instituição financeira de grande porte. O desenvolvimento de software encontra-se descentralizado dentro da empresa, sendo cada departamento responsável pelo processo de desenvolvimento de software a ser adotado. O software avaliado é um sistema legado, foi implantado a cerca de seis anos e possui alto índice de manutenção.

Pode-se classificar o ciclo de vida do sistema analisado como o Modelo em Cascata (SOMERVILLE, 2004), conforme exibido na Figura 19, em que as atividades de

manutenção, geram novas demandas para as atividades de definição de requisitos, projeto, implementação, integração e teste.

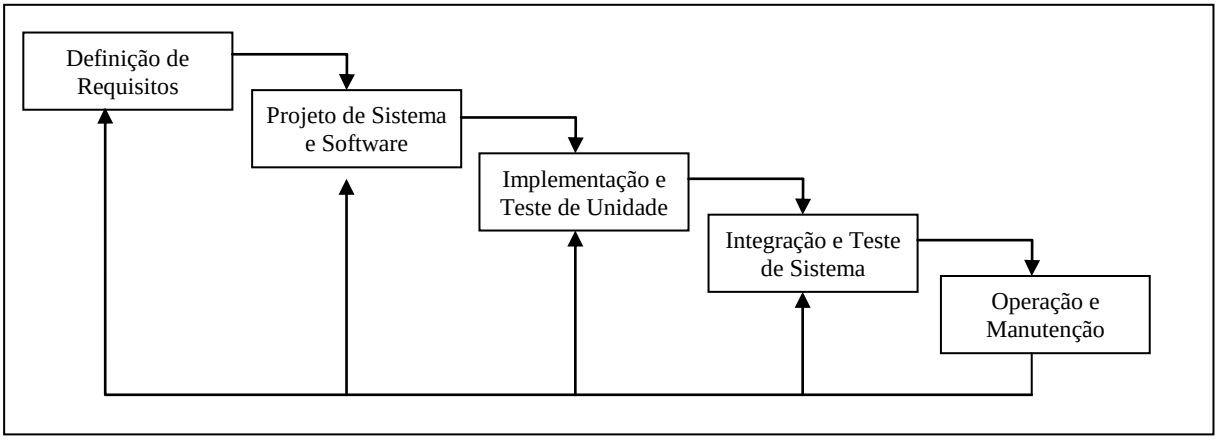

Figura 19 - Modelo Cascata de desenvolvimento de software (SOMMERVILLE, 2004)

As atividades de manutenção geradas para o sistema que está sendo analisado podem se referir a:

- Alterações nas regras a negócio;
- Mudanças na legislação;
- Adaptações a novas tecnologias, podendo envolver hardware ou software;
- Correção de erros criados durante o desenvolvimento do sistema ou em manutenções anteriores;
- Manutenções evolutivas, buscando aumentar a confiabilidade do sistema.

A Figura 20 demonstra o total de requisições de manutenções geradas durante o ano de 2007, e a quantidade de requisições por período de trabalho gasto em cada requisição. Para o total de horas gastas em cada requisição, foram computados os totais de horas gastas em análise, especificação de requisitos e testes funcionais.

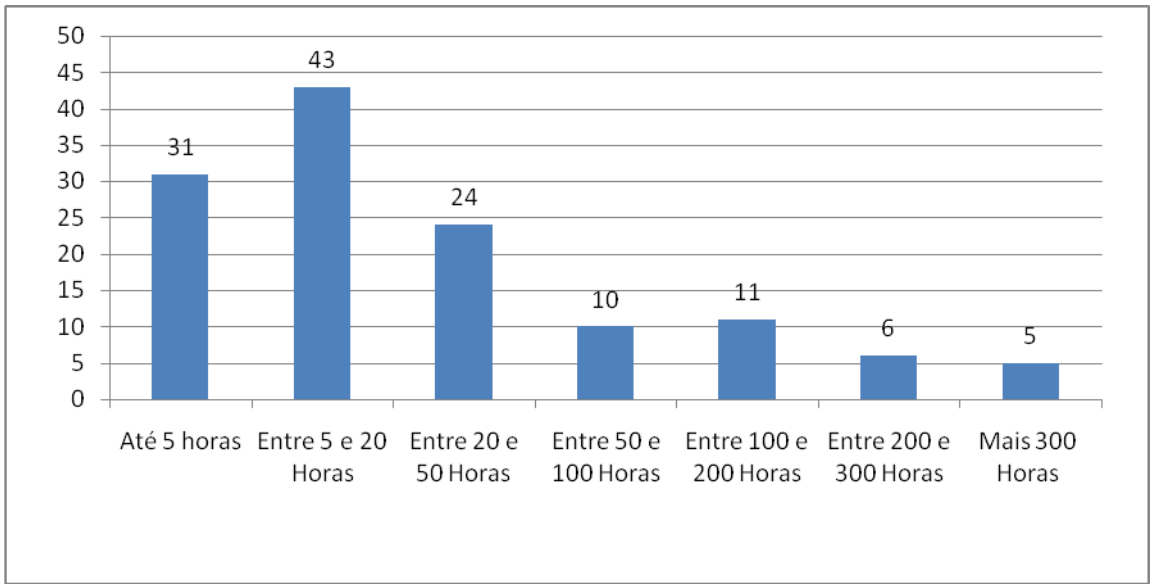

Figura 20 - Quantidade de Requisições por Total de Horas Trabalhadas em 2007

5.3 Preparação do Estudo de Caso

Para a realização do estudo de caso foram selecionadas seis requisições de manutenção. Para a análise inicial foram selecionados três projetos que não utilizaram um processo de teste de software estruturado e para a análise posterior foram selecionados outros três projetos que tiveram o processo de teste aplicado.

A análise inicial utilizou um projeto de curto prazo (18 horas), um projeto de médio prazo (53 horas) e um projeto de maior duração (117 horas), identificados respectivamente por S1, S2 e S3. O pré-requisito para a seleção dos projetos era que eles possuíssem a documentação de requisitos baseada em casos de uso, porém como não possuíam um processo de teste estruturado não foram gerados todos os artefatos de teste necessários.

A análise realizada após a implantação do processo de testes utilizou um projeto de curto prazo (29 horas), um projeto de médio prazo (62 horas) e um projeto de maior duração (240 horas), identificados respectivamente por S4, S5 e S6. Foram selecionados projetos que possuíssem a documentação de requisitos com cronograma atualizado e artefatos de teste propostos para a execução dos testes funcionais de software.

A equipe selecionada para a realização das inspeções envolveu quatro pessoas que se revezaram nos papéis propostos: autor, moderador, inspetor, leitor e escritor. As pessoas já faziam parte da equipe de desenvolvimento da empresa, possuíam formação na área de tecnologia da informação e diferentes graus de experiência.

Foi realizado um treinamento de 3 horas com os inspetores para a explicação do funcionamento dos processos, sua aplicação e os objetivos do estudo de caso. Foram apresentados os artefatos que deveriam criados e a forma correta para o seu preenchimento.

5.4 Reuniões de Inspeção

A Tabela 18 demonstra os resultados obtidos na inspeção inicial, em que não havia um processo de teste estruturado. Os artefatos utilizados para a documentação dos testes funcionais não eram os mesmos que os utilizados após a implantação do processo de teste estruturado, porém devido à semelhança das informações contidas neles foi possível realizar a inspeção.

Os artefatos foram distribuídos antecipadamente aos participantes e devido à diferença na experiência dos envolvidos com a execução das inspeções, houve divergência quanto aos defeitos identificados individualmente. Durante as reuniões foram excluídos os defeitos identificados erroneamente na preparação individual

Solicitação de Manutenção	Identificação Artefato	Artefato	Defeitos Identificados	Nota Avaliação	Status
S1	01	Casos de Testes	4	4,2	Regular
S1	02	Procedimentos de Testes	5	5,4	Regular
S2	03	Casos de Testes	5	2,9	Ruim
S2	04	Casos de Testes	3	5,7	Regular
S2	05	Procedimentos de Testes	4	6,3	Bom
S2	06	Procedimentos de Testes	4	6,3	Bom
S3	07	Casos de Testes	4	4,2	Regular
S3	08	Casos de Testes	5	5,7	Regular
S3	09	Casos de Testes	2	5,7	Regular
S3	10	Procedimentos de Testes	6	4,5	Regular
S3	11	Procedimentos de Testes	4	6,3	Bom
S3	12	Procedimentos de Testes	5	5,5	Regular

Tabela 18 – Resultados da Inspeção Inicial

Foram analisados 12 artefatos, sendo um deles classificado como “Ruim” (8%), 8 como “Regular” (67%) e 3 obtiveram o conceito “Bom” (25%), conforme critérios detalhados no capítulo 4.

A Tabela 19 demonstra os resultados obtidos na inspeção aplicada após a implantação do processo de testes. Os artefatos foram distribuídos antecipadamente aos participantes e durante as reuniões os defeitos previamente identificados, foram reavaliados.

Solicitação de Manutenção	Identificação Artefato	Artefato	Defeitos Identificados	Nota Avaliação	Status
S4	PL-01	Plano de Teste	4	8	Bom
S4	CT-01	Casos de Teste	2	7,1	Bom
S4	PR-01	Procedimento de Testes	3	7,2	Bom
S4	IT-01	Incidente de Testes	2	6,7	Bom
S5	PL-01	Plano de Teste	5	7	Bom
S5	CT-01	Casos de Teste	2	7,1	Bom
S5	CT-02	Casos de Teste	2	7,1	Bom
S5	CT-03	Casos de Teste	2	7,1	Bom
S5	PR-01	Procedimento de Testes	3	7,2	Bom
S5	PR-02	Procedimento de Testes	0	10	Ótimo
S5	PR-03	Procedimento de Testes	1	9	Ótimo
S5	IT-01	Incidente de Testes	2	6,7	Regular
S5	IT-02	Incidente de Testes	1	8,3	Ótimo
S5	RT-01	Resumo de Testes	1	8	Bom
S6	PL-01	Plano de Teste	3	8,3	Ótimo
S6	CT-01	Casos de Teste	1	8,6	Ótimo
S6	CT-02	Casos de Teste	3	5,7	Bom
S6	CT-03	Casos de Teste	2	7,1	Bom
S6	CT-04	Casos de Teste	2	7,1	Bom
S6	CT-05	Casos de Teste	1	8,6	Ótimo
S6	PR-01	Procedimento de Testes	2	8,1	Ótimo
S6	PR-02	Procedimento de Testes	3	7,2	Bom
S6	PR-03	Procedimento de Testes	2	8,1	Ótimo
S6	PR-04	Procedimento de Testes	0	10	Ótimo
S6	PR-05	Procedimento de Testes	3	7,2	Bom
S6	IT-01	Incidente de Testes	1	8,3	Ótimo
S6	IT-02	Incidente de Testes	2	6,7	Regular
S6	RT-01	Resumo de Testes	1	8	Bom

Tabela 19 – Resultados da Inspeção após a Implantação do Processo de Testes

Foram analisados 28 artefatos, sendo 2 deles classificados como “Regular” (7%), 16 como “Bom” (57%) e 10 obtiveram o conceito “Ótimo” (36%), conforme critérios detalhados no capítulo 4. A Figura 21 apresenta a classificação dos artefatos inspecionados.

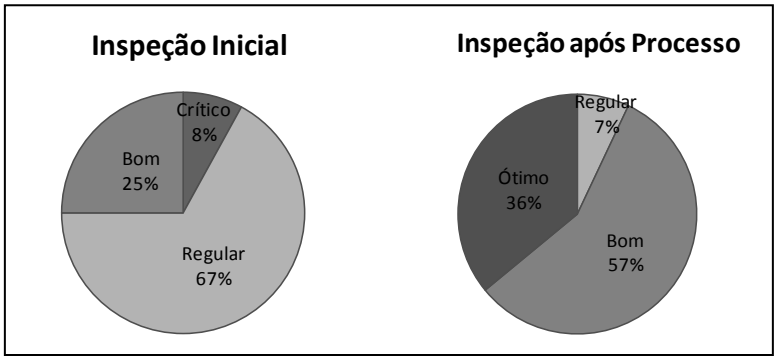

Figura 21 - Classificação dos Artefatos Inspecionados

Após a realização das reuniões de inspeção e a identificação dos defeitos, os artefatos foram corrigidos.

5.5 Considerações Finais

Neste capítulo foi realizado um comparativo, entre as listas de verificação do processo de inspeção proposto neste trabalho e as listas de verificação identificadas em processos de desenvolvimento de software existentes na indústria de software.

O capítulo apresentou também um estudo de caso da aplicação do processo de teste proposto e da inspeção nos artefatos de testes gerados, além dos resultados obtidos com a aplicação dos processos.

O próximo capítulo apresenta as considerações finais sobre o trabalho desenvolvido, suas principais contribuições e limitações, bem como sugestões de trabalhos a serem desenvolvidos.

6 CONSIDERAÇÕES FINAIS

Este capítulo apresenta as considerações finais sobre o trabalho desenvolvido, suas principais contribuições, limitações, e sugestões de trabalhos a serem desenvolvidos.

A necessidade de desenvolvimento de software com qualidade, dentro de prazos e custos estimados e com baixo índice de defeitos, faz com que os processos de desenvolvimento sejam revistos e melhorados. O objetivo das inspeções de software é melhorar a qualidade de artefatos de software através de sua análise, detectando e removendo defeitos antes que o artefato seja enviado para as etapas seguintes do processo de desenvolvimento de software. Os objetivos do teste de software são a execução de um programa para encontrar erros e a validação dos requisitos definidos pelo usuário. Porém, avaliar se todos os requisitos foram atingidos completamente é um objetivo muito difícil de ser atingido, pois depende de interpretação individual.

Dentro deste contexto, o objetivo deste trabalho foi o de propor um processo de inspeção em artefatos de teste de software, amparado por um processo de testes funcionais de software. Durante a execução do trabalho foram definidas as atividades que devem ser realizadas no processo de inspeção e os respectivos responsáveis. Foram propostos os artefatos que serão utilizados durante os processos de teste e inspeção, além das listas de verificação criadas como ferramenta auxiliar para a execução das reuniões de inspeção. Junto com as listas de verificação foram definidos os critérios de avaliação dos artefatos inspecionados, possibilitando classificá-los como artefatos aprovados ou reprovados.

Para a validação da proposta foi realizado um estudo de caso em projetos desenvolvidos por uma equipe de desenvolvimento de software em uma instituição financeira, além de um comparativo das listas de verificação propostas e listas disponíveis na indústria de software.

Uma das limitações observadas durante a execução do processo de inspeção foi a pouca experiência por parte das pessoas da equipe na execução das atividades de inspeção, resultando em uma diferença com relação à quantidade de defeitos encontrados individualmente.

O comparativo entre as listas de verificações propostas com as existentes no RUP e OpenUP permitiu identificar alguns benefícios no modelo proposto. As listas de verificação propostas atingem um número maior de artefatos de teste em relação aos outros dois modelos. Além disso, os critérios propostos para a avaliação de aprovação dos artefatos não fazem parte das listas de verificação comparadas.

A aplicação do processo de inspeção juntamente com um processo de teste claramente definido permitiu observar, em projetos reais, a melhora na qualidade dos artefatos desenvolvidos, e o uso das listas de verificação, possibilitou a identificação e correção das não-conformidades existentes nos artefatos de teste gerados.

6.1 Sugestões de Trabalhos Futuros

Os seguintes trabalhos podem ser sugeridos como forma de extensão do trabalho desenvolvido:

- Replicação: devem ser realizados novos experimentos de replicação para a confirmação do processo de inspeção, aplicando o processo também em artefatos de testes criados para outras técnicas de testes, além dos testes funcionais;
- Ferramenta de apoio: construção de uma ferramenta de apoio para melhorar a produtividade e qualidade das atividades relacionadas ao processo de inspeção em artefatos de teste de software;
- Métricas de avaliação: propor novas métricas de avaliação para uma nova validação da conformidade do processo de inspeção proposto. Propor métricas que possam avaliar os aspectos referentes a custo e prazo
- Técnica de leitura: desenvolvimento de uma técnica de leitura específica para artefatos de teste de software;
- Métodos ágeis de desenvolvimento: aplicar o processo de inspeção proposto em equipes que trabalham com métodos ágeis de desenvolvimento.

REFERÊNCIAS BIBLIOGRÁFICAS

AHERN, D.; CLOUSE, A.; TURNER, R. **CMMI Distilled: A Practical Introduction to Integrated Process Improvement,** SEI Series in Software Engineering, Addison- Wesley, 2001.

AURUM, A., PETERSSON, H., WOHLIN, C. **State-of-the-Art: Software Inspections after 25 Years**. Software Testing Verification and Reliability, Vol. 12, No. 3, pp. 133-154, 2002.

BASILI, V., *et al*. **The Empirical Investigation of Perspective-Based Reading**. Empirical Software Engineering: An International Journal, 1996.

BASILI, V. **Evolving and Packaging Reading Technologies.** The Journal of Systems and Software, 1997.

BERLING, T., THELIN, T. **A case study of reading techniques in a software company**. International Symposium on Empirical Software Engineering, 2004.

BOEHM, W., BASILI, V. **Software Defect Reduction Top 10 List**. IEEE Computer, v. 34, n. 1, pp. 135-137, 2001.

CIOLKOWSKI, M. **Software Inspections, Reviews & Walkthroughs**. Proceedings of the 24th International Conference on Software Engineering, ACM Press, 2002.

CMMI. **CMMI for Development, Version 1.2, Improving processes for better products**, Carnegie Mellon University - SEI Software Engineering Institute, August 2006. Disponível em <http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html>. Acesso em 10/10/2006.

COLLOFELLO, J. **The Software Technical Review Process**. Carnegie Mellon University - SEI Software Engineering Institute, June 1988. Disponível em <ftp://ftp.sei.cmu.edu/pub/education/cm3.pdf >. Acesso em 10/06/2007.

DELAMARO, M., MALDONADO, J., JINO, M. **Introdução ao teste de software.** Elsevier, 2007.

DENGER, C., SHULL, F. **A Practical Approach for Quality-Driven Inspections.** IEEE Software, v.24, n.2, pp. 79-86, 2007.

DORIA, E. **Replicação de estudos empíricos em engenharia de software**. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação - ICMC-USP, São Carlos, 2001.

FAGAN, M. **Design and code inspection to reduce Errors in Program Development.** IBM Systems Journal, v. 15, n. 3, 1976.

GREGOLIN, R. **Uma proposta de inspeção em modelos de caso de uso**. Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo, São Paulo, 2007.

HATTON, L. **Empirical Test Observations in Client-Server Systems**. IEEE Computer Society, v.40, n.5, pp. 24-29, 2007.

IEEE 610. **IEEE 610 Standard Glossary of Software Engineering Terminology**. IEEE Institute of Electrical and Eletronics Engineers - Computer Society, 1990.

IEEE 829. **IEEE 829 Standard for Software Test Documentation.** IEEE Institute of Electrical and Eletronics Engineers - Computer Society, 1998.

IEEE 1012. **IEEE 1012 Standard for Software Verification and Validation.** IEEE Institute of Electrical and Eletronics Engineers - Computer Society, 1998.

IEEE 1028**. IEEE 1028 Standard for Software Reviews**. IEEE Institute of Electrical and Eletronics Engineers - Computer Society, 1997.

ISO. **ISO/IEC 12119. Information Tecnology - Software Packages - Quality Requeriments and Testing**. 1994.

ISO. **ISO/IEC 15504-5:2003.Information Technology. Process Assessment – Part 5: An Exemplar Process Assessment Model**. 2003.

LAITENBERGER, O., *et al*. **An Experimental Comparison of Reading Techniques for Defect Detection in UML Design Documents.** Journal of Systems and Software, ACM, 2000.

MYERS, G. **The Art of Software Testing.** John Wiley &Sons, 2nd ed., 2004.

OPENUP. **Open Unified Process.** Version 1.5, 2008. Disponível em: <http://www.epfwiki.net/wikis/openup/index.htm>. Acesso em 01/06/2008.

PARNAS, D., LAWFORD, M. **The Role of Inspection in Software Quality Assurance.** IEEE Transactions On Software Engineering, Vol. 29, No. 8, 2003.

PAULK, M., *et al*. **The Capability Maturity Model*:* Guidelines for Improving the Software Process**, Addison Wesley, 1999.

PORTER, A., VOTTA, L., BASILI, V. **Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment**. lEEE Transactions on Software Engineering, Volume 21, Issue 6, 1995.

PRESSMAN, R. **Software Engineering a Practioner´s Approach.** Mc Graw Hill, 5a. edition, 2001.

ROCHA, A.; MALDONADO J.; WEBER K. **Qualidade de Software Teoria e Prática**. Prentice Hall, 2001.

RUP. **Rational Unified Process.** Version 2003.06.15, 2005. Disponível em: <http://www.ts.mah.se/RUP/RationalUnifiedProcess>. Acesso em 01/06/2008.

SHULL, F. **Developing Techniques for Using Software Documents: A Series of Empirical Studies**. PhD Thesis, Department of Computer Science, University of Maryland, USA, 1998.

THELIN, T., RUNESON, P., WOHLIN, C. **An experimental comparison of usage-based and checklist based reading.** Proceedings of the 10th International Symposium on Software Metrics (METRICS'04), 2003.

THELIN, T., *et al*. **Evaluation of Usage-Based Reading-Conclusions after Three after Three Experiments**. Empirical Software Engineering, Volume 9, Issue 1-2, 2004.

TRAVASSOS, G., *et al*. **Detecting defects in object-oriented designs: using reading techniques to increase software quality**. Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, Volume 34, Issue 10, 1999.

WINKLER, D., RIEDL, B., BIFFL, S. **Improvement of design specifications with inspection and testing.** Software Engineering and Advanced Applications, 31st EUROMICRO Conference, 2005.

APÊNDICE A – ARTEFATOS DE TESTE E INSPEÇÃO

A.1 Plano de Testes

Plano de Teste

NNNN - XXXXXXXXXXXXXXXXXXX

<NNNN: identificador –XXXXXX: descrição do plano de testes >

Projeto: XX – Nome do Projeto

Versão do Documento

Versão	*Data*	*Comentários*	*Autor*

1. Introdução

[Cada uma das seções desse documento contém tópicos para inclusão em uma estratégia de teste. O texto disponível dentro de cada uma das subseções serve para orientação e pode ser excluído, alterado e/ou adicionado.]

1.1 Propósito

[Parágrafo introdutório descrevendo o propósito desse documento. Exemplo: "Esse documento descreve o escopo, abordagem, recursos e cronograma das atividades de teste para o teste XYZ1234. Identifica os itens que estão sendo testados, características a serem testadas, as tarefas de teste devem ser desempenhadas, as pessoas responsáveis por cada tarefa e os riscos associados a essa estratégia."]

1.2 Escopo

[Descreva os estágios do teste por exemplo, unidade, integração ou sistema e os tipos de teste que serão endereçados por este plano, como, por exemplo, Funcional e de Performance. Liste quaisquer riscos ou contingências que podem afetar que possam impactar o projeto, desenvolvimento ou implementação dos testes.]

2. Documentações do Projeto

A tabela abaixo identifica a documentação utilizada e sua disponibilidade para desenvolver o plano de teste.

Documento	Criado ou Disponível	Aceito ou Revisado	Observações
Especificações dos Requisitos	☐ Sim ☐ Não	☐ Sim ☐ Não	
Especificações dos Casos de Uso	☐ Sim ☐ Não	☐ Sim ☐ Não	
Especificação/Modelo de Projeto	☐ Sim ☐ Não	☐ Sim ☐ Não	
Cronograma do projeto	☐ Sim ☐ Não	☐ Sim ☐ Não	

3. Contexto do Projeto

[Entre com uma descrição breve da aplicação em teste (componentes, aplicações, sistemas, etc.) e seus objetivos. Inclua informações tais como funções importantes, arquiteturas, e um breve histórico do projeto, bem como informações relacionadas à arquitetura onde as aplicações serão executadas.]

3.1 Sistemas/Áreas Envolvidos

[Providenciar um diagrama que detalhe os sistemas e áreas envolvidas dentro dessa estratégia de teste.]

3.2 Interfaces

[Identificar todas as interfaces com outros sistemas e negócios que devem ser testados. Incluir alguma interface entre esse projeto e algum outro projeto atualmente sendo desenvolvido.]

4. Funcionalidades a Serem Testadas

[Identificar as características e combinações do sistema a ser testado, como por exemplo:

- *Interfaces (Externas e Internas se necessário)*
- *Funcionalidades de telas*
- *Simulação de processamento do Negócio*

- *Backup/ Restore*
- *Recuperação de Falhas*
- *Desempenho/ Volume/ Limites/ Esgotamentos*
- *Operação/ Manutenção e Limpezas/ Administração Base de Dados*
- *Processamentos regulares*
- *Controles de Segurança e Acesso (ex: Teste de Integridade de Base de Dados, usabilidade e acessibilidade)*
- *Auditoria*
- *Treinamento*
- *Instalação e Implantação/ Distribuição do Sistema]*

5. Funcionalidades a Não Serem Testadas

[Identificar as funcionalidades que não serão testadas no plano de testes elaborado e a justificativa]

6. Itens de Teste

Identificar os itens a serem testados, baseados no documento de Especificação de Requisitos de Software, onde os mesmos foram identificados com ID Requisito como referência para garantir a integridade do teste.

ID Requisito	Descrição Requisito
1	
2	
3	
4	
5	

7. Critério de Aceite / Falha dos Itens de Teste

Critérios especiais podem ser determinados se cada item do teste passou ou falhou no teste. Esse critério deve ser acumulativo, da passagem de um item para outro e para evolução de fases dos testes. *[Exemplo: todos os itens do teste passarão sem falhas.]*

ID Requisito	Descrição Requisito	Critério Aceite / Falha
1		
2		
3		
4		
5		

8. Estratégia de Testes

[A estratégia de teste apresenta a maneira recomendada de testar a aplicação em teste.
Para cada tipo de teste, forneça uma descrição do teste e o porquê está sendo implementado e executado este teste.
Se um tipo de teste não for implementado e executado, indique isto numa sentença declarada justificando, como "Este teste não será implementado ou executado, por não ser apropriado".
As principais considerações para a estratégia de teste são as técnicas utilizadas e os critérios de finalização para cada tipo de teste.
A estratégia deve ser descrita com detalhes suficientes para permitir a identificação das principais tarefas do teste e estimativas de tempo para execução de cada uma delas.

Identificar restrições significantes no teste tais como disponibilidade do item a ser testado recursos para o teste disponível e data limite.]

8.1 Tipos de Teste

[Abaixo segue exemplo de definição de tipo de testes]

8.1.1 Testes Funcionais

Testes de funcionais devem focar-se em todo requisito de teste que foi derivado diretamente de casos de uso ou funções de negócio e regras de negócio. Este tipo de teste é baseado em técnicas de teste de caixa preta, isto é, verificar a aplicação e processos internos pela interação com a aplicação através da interface gráfica de usuário e análise dos resultados de saída.

Objetivo do Teste:	*Garantir funcionalidade adequada ao software, incluindo navegação, entrada de dados, processamento e recuperação de dados.*
Técnica:	*Executar cada caminho dos casos de uso, ou funções, utilizando dados válidos ou inválidos para verificar o seguinte:* • *O resultado esperado ocorre quando um dado válido é utilizado.* • *O erro apropriado ou mensagem de advertência são exibidos quando dados inválidos são utilizados.* • *Cada regra de negócio está propriamente aplicada.*
Critério de finalização:	• *Todos os testes planejados foram executados.* • *Todo defeito identificado foi repassado aos responsáveis para resolução.*
Considerações especiais:	*Identificação e/ou descrição os itens (internos ou externos) que impactam na implementação e execução dos testes funcionais.*

9. Especificação das Técnicas e Ferramentas

Especificar ferramentas particulares ou técnicas que serão usadas para a execução do Plano de Teste.

	Ferramenta	**Fornecedor**	**Versão**
Gerenciamento de Teste			
Trilha de defeitos			
Ferramenta para testes funcionais			
Ferramenta para testes de performance			
Monitor de cobertura de testes ou perfil			
Gerência de Projetos			
Ferramentas de gerenciamento de Banco de Dados			

10. Artefatos de Teste

Identificar os artefatos entregues durante a execução dos testes.

Tarefa	Esforço	Data de início	Data de fim
Plano de Teste			
Projeto de Teste			
Implementação do Teste			
Execução do Teste			
Avaliação do Teste			

11. Atividades do Teste

[Identificar o conjunto de tarefas necessárias para preparar a execução do Projeto de Teste. A seguir, encontram-se os tipos de tarefas que podem ser consideradas:

- *Planejamento do Teste;*
- *Estimativa e cronograma do teste;*
- *Alocação e atribuição de pessoas para o teste;*
- *Análise e Design do Teste;*
- *Especificação do Teste;*
- *Preparação dos dados do Teste;*
- *Execução do Teste;*
- *Gerenciamento do Teste*
- *Apresentação de Aceite do Teste para o Negócio;*
- *Planejamento e Coordenação do ambiente de teste;*
- *Gerenciamento dos incidentes dos testes.]*

12. Necessidades de Ambiente

A seguinte tabela expõe os recursos de sistema necessários para testar o projeto:

Recurso	Descrição
Servidor de banco de dados	
—Rede	
—Nome do servidor	
—Nome do Banco de Dados	
Sistema Operacional	
Segurança e acessos requeridos para área e equipamentos do Teste	
Repositório de Testes	
—Rede	
—Nome do Servidor	
PCs de desenvolvimento dos testes	

13. Equipe de Testes

Esta seção apresenta os recursos recomendados para um projeto, suas principais responsabilidades, e seus conhecimentos e conjunto de habilidades. Se necessário, identificar treinamentos necessários para suprir conhecimentos especiais.

Profissional	Recurso Mínimo Recomendado	Responsabilidades Específicas/ Comentários
Gerente dos testes		Gerenciamento das atividades. Responsabilidades: • Fornece direcionamento técnico • Adquire recursos apropriados • Fornece gerenciamento de relatórios
Projetista de testes		Identifica, prioriza, e implementa casos de teste. Responsabilidades: • Gerar planos de teste • Gerar modelo de teste • Avaliar efetividade do esforço de teste
Testador		Executar os testes. Responsabilidades: • Executar testes • Gravar resultados • Recuperar erros • Documentar necessidades de mudança
Administrador do sistema		Certificar que o ambiente e ferramentas estão sendo gerenciadas e em manutenção constante. Responsabilidades: • Administrar sistema de gerenciamento de testes • Instalar e gerenciar acesso aos sistemas de teste
DBA e analista de dados		Certificar que os dados de teste e ambiente de banco de dados estão sendo gerenciados e em manutenção. Responsabilidades: • Administrar dados de teste
Desenhista da aplicação		Identificar e definir as operações, atributos, e associações de classes de teste. Responsabilidades: • Identificar e definir componentes e pacotes de testes
Implementador		Implementar os componentes e pacotes de programas de teste Responsabilidades: • Criar os componentes e pacotes de teste para o modelo de teste

14. Riscos e Contingências

[Identificar as os riscos considerados de alta prioridade durante a realização dos do testes.
Podem ser consideradas possibilidades de riscos:

- *Mudança de requerimento de negócios;*
- *Mudança da data final;*
- *Insuficiência de recurso;*
- *Problemas de qualidade;*
- *Necessidade de conhecimentos especiais;*
- *Novas tecnologias]*

A.2 Casos de Teste

Casos de Teste

NNNN - XXXXXXXXXXXXXXXXXX

<NNNN: identificador –XXXXXX: descrição do caso de testes >

Projeto: XX – Nome do Projeto

Versão do Documento

Versão	***Data***	***Comentários***	***Autor***

1 Introdução

1. 1 Propósito

1.2 Escopo

[Especifique o escopo dos Casos de Teste Por exemplo, quais são os testes mais críticos, o que não precisa ser testado ou o que é imprescindível ser testado]

1.3 Referências

[Especifique as ligações destes Casos de Teste com : Casos de Uso e Plano de Testes]

Identificador	Descrição

2. Casos de Teste

2.1 Nome do Caso de Teste

[Especifique de maneira clara, o nome do Caso de Teste. Exemplo: Transferência de Valores].

2.2 Condições para os testes

[Especifique os cenários e para cada um deles, demonstre as características e condições existentes para os testes. Exemplo:

Cenário 1 – Transferência entre contas corrente

Cenário 2 – Transferência entre conta corrente e conta poupança.]

3. Fluxo Básico – Cenário n

3.1 Descrição sucinta do Fluxo Básico

[Entenda Fluxo Básico como o caminho normal da aplicação, referente ao cenário que está sendo testado.]

3.2 Condições a serem testadas

[Especifique detalhadamente todas as condições a serem testadas, as entradas, as ações, as saídas esperadas, enfim, descreva de forma completa o que deve ser contemplado nos testes.]

4. Fluxos Alternativos – Cenário n

4.1 Descrição sucinta do Fluxo Alternativo

[Entenda Fluxos Alternativos como os outros caminhos que a aplicação pode executar, como por exemplo, condições de erro ou situações incomuns previstas .]

4.2 Condições a serem testadas

[Especifique detalhadamente todas as condições a serem testadas, as entradas, as ações, as saídas esperadas, enfim, descreva de forma completa o que deve ser contemplado nos testes.]

A.3 Procedimentos de Teste

Procedimentos de Teste

NNNN - XXXXXXXXXXXXXXXXXXX

<NNNN: identificador –XXXXXX: descrição do procedimento de testes >

Projeto: XX – Nome do Projeto

Versão do Documento

Versão	*Data*	*Comentários*	*Autor*

1. Introdução

1.1 Identificação

[Especifique uma identificação única e um nome para o documento de Procedimentos de Teste que está sendo criado]

1.2 Escopo

[Especifique o escopo do documento de Procedimentos de Testes indicando os passos necessários para executar um determinado conjunto de casos de testes.]

1.3 Referências

[Especifique as ligações deste Procedimento de Testes com : Casos de Uso, Plano de Testes e Casos de Teste.]

1.4 Requisitos Especiais

[Caso seja necessário algum requisito especial durante a execução dos testes, como pré-requisitos, ou ambiente diferenciado, detalhar a necessidade.]

2. Procedimentos de Testes

2.1 Nome do Procedimento de Teste

[Especifique o nome do Procedimento de Testes. É interessante que tenha um nome relacionado ao Caso de Teste relacionado]

2.2 Passos/Ações

[Especifique as instruções de maneira sucinta, indicando os passos a serem tomadas pelo testador, quando estiver executando o Caso de Teste relacionado.]

2.3 Valores de Entrada/Caso de Teste

[Especifique os valores válidos de entrada para o Procedimento de Teste descrito, e que não foi alvo de descrição no Caso de Teste relacionado.]

2.4 Resultado(s) esperado(s)

[Especifique a resposta esperada da aplicação a cada passo/ação. Este resultado pode ser representado pela exibição de uma janela, habilitação de um botão, o resultado de um cálculo ou ainda o retorno de um registro.]

2.5 Métodos para verificação

[Especifique as técnicas para comparar os resultados atuais e esperados, gerados no Caso de Teste relacionado ou no Procedimento de Teste descrito. Por exemplo, para verificar se um requisito foi atendido ou não, executar uma determinada consulta no banco de dados.]

A.4 Resultado de Incidente de Testes

Relatório de Incidentes de Testes

NNNN - XXXXXXXXXXXXXXXXXX

<NNNN: identificador –XXXXXX: descrição do incidente de testes >

Projeto: XX – Nome do Projeto

Versão do Documento

Versão	*Data*	*Comentários*	*Autor*

1. Introdução

1.1 Propósito

1.2 Escopo

[Especifique a abrangência do Teste. Por exemplo, informar que este teste abrangeu apenas funcionalidade ou se foi um teste completo.

1.3 Referências

[Especifique as ligações deste Relatório de Incidente de Testes com: Caso de Uso, Plano de Teste, Casos de Teste e Procedimentos de Teste.]

2. Relatório de Incidente de Testes

2.1 Identificação do Relatório de Incidente de Teste

[Crie um identificador e uma descrição para o Relatório de Incidentes de Testes, exemplo: NNNN – XXXXXXXXXX, onde NNNN é o identificador e XXXXXXXXXX é a descrição .]

2.2 Data, hora, nome do testador e informações do ambiente

[Especifique as informações solicitadas. Sobre informações do ambiente, deve ser descrito o sistema operacional utilizado, características da máquina e outras informações que possam ser importantes para interpretação do Resultado dos Testes.]

2.3 Identificação específica da aplicação em testes

[Especifique as características da aplicação como versão, objetos, componentes relacionados, arquivos utilizados.]

2.4 Casos de Testes relacionados

[Especifique os Casos de Testes que foram cobertos pelos testes e quais os requisitos foram atingidos nos mesmos.]

2.5 Especificações particulares do tipo de teste

[Especifique os resultados de acordo com o tipo de teste efetuado.
Por exemplo :
Testes funcionais – Requisitos atingidos, resultados ou comportamentos inesperados.]

A.5 Relatório de Resumo de Testes

Relatório de Resumo de Testes

NNNN - XXXXXXXXXXXXXXXXXXX

<NNNN: identificador –XXXXXX: descrição do resumo de testes >

Projeto: XX – Nome do Projeto

Versão do Documento

Versão	*Data*	*Comentários*	*Autor*

1. Introdução

1.1 Propósito

1.2 Escopo

[Uma breve descrição do escopo deste Relatório de Resumo de Testes, a qual projeto está associado e algo a mais que possa ser afetado ou influenciado por este documento.]

1.3 Referências

[Especifique as ligações deste Resumo de Avaliação dos Testes com :Casos de Uso, Plano de Teste,Casos de Teste, Procedimentos de Teste e Relatórios de Incidentes de Testes]

2. Sumário dos Resultados dos Testes

[Uma breve descrição dos resultados dos testes.]

3. Cobertura dos testes quanto a requisitos / funcionalidade.

[Especifique se todos os requisitos inicialmente definidos foram atendidos pelos testes.]

4. Ações sugeridas

[Especifique ações a serem tomadas baseadas nas avaliações dos resultados dos testes e métricas pré-estabelecidas.]

5. Pareceres de Projetistas/Testadores

[Projetistas e/ou Testadores devem registrar seus pareceres com relação aos testes, métricas utilizadas, para inclusive servirem de subsídio para futuras manutenções.]

A.6 Registro de Inspeção

Registro de Inspeção

NNNN - XXXXXXXXXXXXXXXXXXX

<NNNN: identificador –XXXXXX: descrição do registro de inspeção >

Projeto: XX – Nome do Projeto

Versão do Documento

Versão	*Data*	*Comentários*	*Autor*

1. Introdução

[O Registro de Inspeção é um artefato para registrar decisões e ações definidas, para referência futura. Também são úteis para comunicação com os envolvidos no projeto, que não participaram da inspeção.]

2. Objetivo da Inspeção

[Identifique o objetivo da inspeção. A inspeção pode ser focada num produto de trabalho específico, ou avaliar o projeto como um todo, num determinado marco, bem como avaliar um subconjunto de produtos de trabalho.]

3. Participantes da Inspeção

[Identifique os participantes da inspeção e seus papéis e responsabilidades]

4. Local e Data

[Informe o local e data da reunião de inspeção.]

5. Decisões

[Informe quais decisões foram tomadas ao longo da inspeção.]

6. Não-Conformidades Encontradas

[Informe as não-conformidades encontradas, classificando por tipo e severidade. Informar em qual artefato foi encontrado o problema e faça uma breve descrição.]

ID	**Tipo** (**F**)altando/ (**E**)rrada/ e(**X**)tra/ (**U**)sabilidade/ (**P**)erformance/ (**N**)ão-defeito	**Severidade** (**P**)rincipal/ (**S**)ecundária	**Localização**	**Descrição**
1				
2				
3				

7. Ações definidas

[Informe quais ações definiu-se executar por conta da inspeção. Informe, se possível, o responsável pela ação e data que deve ser efetuada ou concluída.]

A.7 Listas de Verificação para Inspeção de Artefatos de Teste de Software

Lista de Verificação de Artefatos de Teste Funcionais

TESTES FUNCIONAIS: *são testes utilizados para assegurar que o comportamento do sistema atende às especificações de requisitos de sistema, e utiliza as entradas e saídas produzidas pelo sistema, para avaliar seu comportamento.*

Os artefatos de teste gerados serão checados contra as listas de verificação. A qualidade dos documentos é responsabilidade dos autores e não do revisor.

1. PLANO DE TESTES

Avaliação do artefato plano de testes que apresenta o planejamento das atividades de testes, recursos necessários, cronograma, abordagem dos testes e ambiente requerido

S = Aceito. Documentado conforme padrão esperado.
N = Não Aceito. Documentação incompleta/ incorreta/ ambigua.
NA = Não aplicável para esse projeto.

		S N NA
1.1	Foi criado um identificador único para o plano de testes.	S
1.2	O escopo do plano de testes está claramente definido e delimitado.	S
1.3	Os documentos de projeto utilizados para gerar o plano foram corretamente identificados.	S
1.4	Foram identificados os sistemas, áreas e interfaces envolvidas.	S
1.5	As funcionalidades a serem testadas foram identificadas de forma clara e precisa.	S
1.6	As funcionalidades a não serem testadas foram identificadas de forma clara e precisa e foram incluídas as justificativas.	S
1.7	Os itens a serem testados foram listados de forma clara e precisa.	S
1.8	Os requisitos a serem testados estão devidamente identificados de acordo com o Documento de Especificação de Requisitos.	S
1.9	Estão identificados os critérios de aceite/falha para todos os requisitos identificados para teste, de forma clara e precisa e são suficientes e aceitáveis.	S
1.10	A estratégia de teste documenta os tipos de testes a serem implementados e executados.	S
1.11	Os tipos de testes a serem implementados possuem os objetivos, técnicas e critérios de finalização.	S
1.12	O plano de teste identifica os artefatos criados pelas atividades de testes e datas de inicio e fim para criação dos artefatos.	S
1.13	As atividades de testes foram relacionadas, identificando datas de início e fim e esforço necessários.	S
1.14	A lista de atividades é consistente com o processo adotado para os testes.	S

1.15	A lista de atividades é consistente com os aspectos a testar.	S
1.16	Foram identificados os recursos necessários para implementar e executar os testes, incluindo hardware, software e recursos humanos.	S
1.17	O plano de testes identifica os riscos ou contingências que podem afetar ou impactar o esforço de teste.	NA
1.18	O plano de testes está completo, correto e não ambiguo.	S
	Total	**10**

2. CASOS DE TESTE

Avaliação do artefato casos de teste que define os casos de teste, detalhando os dados de teste, a serem utilizados nas condições

S = Aceito. Documentado conforme padrão esperado.
N = Não Aceito. Documentação incompleta/ incorreta/ ambigua.
NA = Não aplicável para esse projeto.

		S N NA
2.1	Foi criado um identificador único para o Caso de Testes.	S
2.2	O escopo do caso de testes está claramente definido e delimitado.	S
2.3	Os documentos utilizados para gerar os casos de teste foram corretamente identificados.	S
2.4	O documento de casos de testes contém o nome do Caso de Testes, relacionado com o Caso de Uso correspondente, permitindo a rastreabilidade entre eles.	N
2.5	Para cada requisito de teste estão descritos os cenários de testes, fluxos básico e fluxos alternativos	N
2.6	O caso de testes contém os procedimentos de teste que serão usados para executar o caso de testes	NA
2.7	Os casos de teste identificados são suficientes para atender as funcionalidades a serem testadas, previstas no plano de testes.	S
	Total	**6,667**

3. PROCEDIMENTOS DE TESTE

Avaliação do artefato procedimentos de teste em que são detalhadado s a execução de cada teste, incluindo as pré-condições e os passos que devem necessariamente ser seguidos.

S = Aceito. Documentado conforme padrão esperado.
N = Não Aceito. Documentação incompleta/ incorreta/ ambigua.
NA = Não aplicável para esse projeto.

		S N NA
3.1	Foi criado um identificador único para o Procedimento de Testes.	S
3.2	O escopo do procedimento de testes está claramente definido e delimitado.	S
3.3	O documento de procedimentos de testes contém o nome do Procedimento de Testes, relacionado com o Caso de Teste correspondente	S

3.4	Estão relacionados os documentos de referência (casos de usos, especificação de requisitos, casos de teste, procedimentos de testes)	S
3.5	Estão definidos os requisitos especiais para a execução do Procedimento de Testes	S
3.6	Foi descrito um fluxo passo a passo, com as atividades a serem executadas para a execução do Procedimento de Testes	S
3.7	O fluxo é detalhado o suficiente permitindo ser executado manualmente, ou convertido em um script de teste.	S
3.8	Estão descritas as entradas esperadas	S
3.9	Estão descritas as saídas esperadas	S
3.10	As entradas e saídas esperadas são suficientes e realistas	S
3.11	Foram definidos os métodos para verificação dos valores esperados de forma clara e precisa	S
	Total	**10**

4. RELATÓRIO DE INCIDENTE DE TESTES

Avaliação do artefato incidente de testes que detalha, para os testes que falharam, o resultado atual comparando-o com o resultado esperado.

S
N
NA

S = Aceito. Documentado conforme padrão esperado.
N = Não Aceito. Documentação incompleta/ incorreta/ ambigua.
NA = Não aplicável para esse projeto.

4.1	Foi criado um identificador único para o relatório de Incidente de Testes.	S
4.2	Estão relacionados os documentos de referência (casos de usos, especificação de requisitos, casos de teste, procedimentos de testes, incidentes de testes).	S
4.3	Estão identificadas a data e hora de realização dos testes.	S
4.4	Estão identificados os nomes do testador e demais pessoas envolvidas no teste.	S
4.5	Estão descritos os ambientes de hardware e software utilizados para a execução dos testes.	S
4.6	Foram identificados os Casos de Testes executados que geraram o documento de Relatório de Incidentes .	S
	Total	**10**

5.0 RELATÓRIO DE RESUMO DE TESTES

Avaliação do artefato resumo de testes que apresenta de forma resumida o resultados dos testes realizados

S
N
NA

S = Aceito. Documentado conforme padrão esperado.
N = Não Aceito. Documentação incompleta/ incorreta/ ambigua.
NA = Não aplicável para esse projeto.

5.1	Foi criado um identificador único para o relatório de Resumo de Testes.	S
5.2	O escopo do resumo de testes está claramente definido e delimitado.	N
5.3	Estão relacionados os documentos de referência (casos de usos, especificação de requisitos, casos de teste, procedimentos de testes, incidentes de testes).	S
5.4	O campo sumário dos resultados resume os incidentes ocorridos, resolvidos ou não.	S
5.5	Estão descritos os requisitos e as funcionalidades efetivamente abrangidas durante a execução dos testes.	S
	Total	**8**

PLANO DE TESTES	**10**
CASOS DE TESTE	**6,667**
PROCEDIMENTOS DE TESTE	**10**
RELATÓRIO DE INCIDENTE DE TESTES	**10**
RELATÓRIO DE RESUMO DE TESTES	**8**

AVALIAÇÃO GERAL ARTEFATOS DE TESTE	**8,9**