

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA - CIn

Pós-Graduação em Ciência da Computação

**Avaliação de Progresso da Qualidade em Projetos de
Software**

Por

Aristides Vicente de Paula Neto

Dissertação de Mestrado

**Recife
Julho / 2011**



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA - CIn

Pós-Graduação em Ciência da Computação

**Avaliação de Progresso da Qualidade em Projetos de
Software**

Por

Aristides Vicente de Paula Neto

Este trabalho foi apresentado à Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: Profº Dr. Hermano Perrelli de Moura

**Recife
Julho 2011**

Ficha Catalográfica

*Este trabalho é dedicado aos meus pais:
Mirian Vicente de Paula e José Regis da Silva*

*“Honra teu Pai e tua Mãe, que é o primeiro
mandamento com promessa;
Para que te vá bem, e vivas muito tempo
sobre a terra.”
Efésios 6:2-3*

*“Porque haverá certamente um futuro e tua
esperança não será frustrada.”
Provérbios 23:18*

Agradecimentos

Dedico meus sinceros agradecimentos

- a Deus, fonte de todo o conhecimento, da sabedoria e da vida eterna, pela sua presença constante na minha vida, pelo auxílio nas minhas escolhas e por realizar os desejos do meu coração. Porque Dele, por Ele e para Ele são todas as coisas.

- ao meu pai, José Régis, e à mãe, Mirian de Paula, pelo amor, carinho, ensinamentos, paciência e confiança dedicada a minha pessoa. Aos meus avos paternos e maternos (*in memoriam*), pelo cuidado, pela alegria e momentos inesquecíveis. Pelo jantares em família, todos sentados a mesa, rindo e brincando. Os momentos passam, ficam as lembranças e as vontades.

- ao meu orientador, Prof^o. Dr. Hermano Perrelli de Moura, pela amizade, confiança e por sua valiosa orientação na condução deste trabalho. Por me apontar os caminhos do conhecimento, pelas palavras de incentivos e pelas orientações presenciais e via *skype*. Pela sua disponibilidade, pela compreensão, pela paciência e pelas dicas. Obrigado!

- ao Prof^o. Dr. Alexandre Vasconcelos e a Prof^a. Dr^a Cristine Gusmão pelo apoio, pela parceria e por terem me aceito junto ao grupo PROMISE (*Process Management and Improvements in Software Engineering*). Pelo compartilhamento de informações e de trabalhos.

- aos grupos de pesquisas, GP2 (*Project Management*) e PROMISE, pelas contribuições, opiniões, compartilhamento de idéias, material de estudo e discussões. Obrigado Alexandre Luna, Francielle Santos, Ivaldir Junior, Humberto Rocha, Júlio Venâncio e Audrey Vasconcelos.

- ao Centro de Informática (CIn) pela excelência e aos professores, por apontarem os caminhos do conhecimento e sempre estarem disponíveis ao desenvolvimento da ciência e do aprendizado.

- aos colegas que encontrei durante este curso de Mestrado em Ciência da Computação do Centro de Informática da Universidade de Pernambuco.

- a Superintendência de Tecnologia de Informação (STI) da Secretária da Fazenda de Pernambuco (SEFAZ-PE) que possibilitou a interação e integração da universidade com a organização. Obrigado Ana Paula Serrano.

- a Gerência de Desenvolvimento de Sistemas (STI-GEDS) pela compreensão, confiança e apoio no trabalho desenvolvido. Obrigado Ana Paula Santos, Luciano Guedes e Sérgio Terra.

- a Gerência de Planejamento e Qualidade (STI-GEPQ) pelas diversas observações, recomendações, orientações e apoio ao meu trabalho. Pelo conhecimento e pelas experiências compartilhadas na área da qualidade de software. Obrigado Renata Kaufman, Marivete Correia, Misol Berenguer, Pedro Paulo e Reinaldo Previtalli.

- a todos da equipe do Cadastro pela confiança e incentivo ao meu desenvolvimento pessoal e profissional. Aos presentes e ausentes. Obrigado Cleto Xavier, Fabio Souza, José Abner, Gustavo Diniz, Rodrigo Simões, José Reinaldo Filho, Jeyvison Andrade, Rafael Câmara, Edgard Neto, Yara Souza, Frederico Lira, Rafael Kayo, Celso Agra Filho, Daniel Kunzler, Juliana Silva e Paulo Burlamaqui.

- a Unisys Brasil pela compreensão e apoio. Por entender que o sucesso de uma organização é, também, o sucesso de seus colaboradores.

- a Ana Carolina Dias, Ivaldir Junior, Ramon Pontes e Silvia Taveiros pela revisão textual do trabalho, sempre me apresentado novos pontos de vista.

- a Roberta Queiroz e Nivia Cruz pelo incentivo e pelos momentos de alegria. Mesmo que a distância e as obrigações da vida nos levem a caminhos diferentes. O carinho e a amizade não têm fim.

- por fim, a todos que contribuíram direta ou indiretamente.

Resumo

Projetos de desenvolvimento de software têm diversos problemas relacionados a custo, prazo e qualidade. Como garantir o sucesso do projeto e a satisfação de clientes e usuários, atendendo todas as suas expectativas com qualidade tem sido uma questão intensamente estudada. O gerente de projeto e toda a equipe são responsáveis pela qualidade e o cumprimento dos objetivos estabelecidos no projeto. Este trabalho propõe um processo de avaliação de progresso da qualidade de projetos de software. O processo proposto pode ser aplicado a projetos baseados no *Rational Unified Process* (RUP), iterativos, incrementais e baseados em casos de uso, e considera o grau de importância dos critérios de qualidade e a complexidade dos casos de uso para cada etapa do projeto.

Um conjunto de atividades foi definido para condução da avaliação de progresso da qualidade e foram estabelecidas métricas de acompanhamento da qualidade durante as etapas do projeto, gerando, assim, uma visão de qualidade. Essa visão de qualidade permite identificar o nível de qualidade do software em cada etapa do seu ciclo de vida. A visão de qualidade será incorporada ao processo de acompanhamento de progresso de projeto de software, o Inspector. Uma ferramenta web foi desenvolvida para servir de apoio às atividades definidas no processo de avaliação de progresso da qualidade em projetos de software.

Palavras-chave: Qualidade de Software, Critérios de Qualidade, Progresso de Qualidade, Métricas de Qualidade.

Abstract

Software development projects have many problems related to cost, schedule and quality. How to ensure project success and satisfaction of clients and users, meeting all your expectations with quality has been an intensely studied issue. The project manager and all staff are responsible for quality and meet the goals established in the project. This dissertation proposes a process for evaluating progress the quality of software projects. The proposed process can be applied to projects based on the Rational Unified Process (RUP), iterative, incremental and based on use cases, and considers the degree of importance of the criteria of quality and complexity of use cases for each milestone.

A set of activities was set to conduct the evaluation of progress and quality metrics were established to monitor the quality during all stages of the project, thus generating a vision of quality. This view allows you to identify the quality level of software quality at every stage of their life cycle. The vision of quality will be incorporated into the process of monitoring the progress of software project, the Inspector. A web tool was developed to provide support for activities defined in the process of evaluating the progress of quality in software projects.

Key-words: Software Quality, Quality Criteria, Quality Progress, Quality Metrics.

Sumário

1. INTRODUÇÃO.....	1
1.1. CONTEXTUALIZAÇÃO	1
1.2. MOTIVAÇÃO	2
1.3. OBJETIVOS DO TRABALHO	3
1.3.1. <i>Objetivo Geral</i>	3
1.3.2. <i>Objetivos Específicos</i>	3
1.4. METODOLOGIA	4
1.5. ESTRUTURAÇÃO DA DISSERTAÇÃO.....	6
2. REFERENCIAL TEÓRICO.....	8
2.1. QUALIDADE DE SOFTWARE.....	8
2.1.1. <i>Conceito de Qualidade</i>	10
2.2. NORMAS E MODELOS DE QUALIDADE	11
2.2.1. <i>CMMI</i>	11
2.2.2. <i>MPS.Br</i>	16
2.2.3. <i>ISO/ IEC 15504</i>	18
2.2.4. <i>ISO/ IEC 9126</i>	20
2.3. AVALIAÇÃO DE PRODUTO DE SOFTWARE	23
2.4. GARANTIA DA QUALIDADE.....	25
2.5. PAPEL DO ANALISTA DE QUALIDADE	27
2.6. PROCESSO INSPECTOR	28
2.7. TRABALHOS RELACIONADOS.....	29
2.8. CONSIDERAÇÕES FINAIS	31
3. O PROCESSO: INSPECTOR Q	33
3.1. CONCEITOS BÁSICOS	33
3.1.1. <i>Fluxo de Atividades</i>	33
3.1.2. <i>Atividades</i>	34
3.1.3. <i>Papéis</i>	34
3.1.4. <i>Responsável</i>	34
3.1.5. <i>Artefatos</i>	34
3.2. VISÃO GERAL DO PROCESSO	35
3.2.1. <i>Papéis</i>	35
3.2.1.1. <i>Gerente de Projetos</i>	36
3.2.1.2. <i>Coletor de Informações</i>	36
3.2.2. <i>Artefatos Produzidos</i>	37
3.2.2.1. <i>Plano de Avaliação</i>	37
3.2.2.2. <i>Critérios de Avaliação da Qualidade dos Artefatos</i>	37
3.2.2.3. <i>Checklist de Avaliação da Qualidade dos Artefatos</i>	38
3.2.2.4. <i>Formulário para Coleta de Informações sobre o Caso de Uso</i>	38
3.2.2.5. <i>Formulário para Coleta de Informações sobre o Projeto</i>	38
3.2.2.6. <i>Relatório de Avaliação de Progresso de Qualidade</i>	38
3.2.3. <i>Fluxo de Atividades</i>	39
3.2.3.1. <i>Planejar avaliação</i>	39
3.2.3.2. <i>Coletar e processar dados da qualidade</i>	40

3.2.3.3.	<i>Avaliar os resultados</i>	41
3.2.3.4.	<i>Solucionar os problemas</i>	42
3.3.	DEFINIÇÃO DE COMPLEXIDADE	43
3.4.	QUALIDADE DE UM CASO DE USO.....	44
3.5.	MÉTRICA DE PROGRESSO DA QUALIDADE	45
3.5.1.	<i>Qualidade dos Artefatos</i>	46
3.5.2.	<i>Qualidade do Caso de Uso</i>	47
3.5.3.	<i>Qualidade do Projeto</i>	47
3.6.	EXEMPLO DA OBTENÇÃO DA VISÃO DE QUALIDADE	48
3.6.1.1.	<i>Primeira Avaliação (Fase Concepção)</i>	50
3.6.1.2.	<i>Segunda Avaliação (Fase Elaboração)</i>	52
3.6.1.3.	<i>Terceira Avaliação (Fase Construção)</i>	53
3.6.1.4.	<i>Avaliação Geral</i>	54
3.7.	FERRAMENTA INSPECTOR PANEL.....	58
3.8.	CONSIDERAÇÕES FINAIS	60
4.	ESTUDO DE CASO	62
4.1.	INTRODUÇÃO	62
4.2.	CARACTERÍSTICAS DO AMBIENTE E DO PROJETO.....	63
4.3.	METODOLOGIA DO ESTUDO DE CASO	64
4.4.	AVALIAÇÃO DO PROGRESSO DA QUALIDADE	66
4.4.1.	<i>Atividades Realizadas</i>	67
4.4.1.1.	<i>Planejar a avaliação</i>	67
4.4.1.2.	<i>Coletar e processar dados da qualidade</i>	68
4.4.1.2.1.	<i>Primeira Avaliação</i>	68
4.4.1.2.2.	<i>Segunda Avaliação</i>	73
4.4.1.2.3.	<i>Terceira Avaliação</i>	77
4.4.1.3.	<i>Avaliar os resultados</i>	82
4.4.1.4.	<i>Solucionar os problemas</i>	85
4.5.	CONSIDERAÇÕES FINAIS	85
5.	CONCLUSÕES	87
5.1.	CONTRIBUIÇÕES	87
5.2.	COMPARATIVO ENTRE OS TRABALHOS RELACIONADOS	89
5.3.	LIMITAÇÕES.....	91
5.4.	TRABALHOS FUTUROS	92
5.5.	CONSIDERAÇÕES FINAIS	93
	REFERÊNCIAS BIBLIOGRÁFICAS	95
	APÊNDICES	102
	ANEXOS	109

Lista de Figuras

Figura 1. Metodologia do Trabalho.....	5
Figura 2. Elementos Fundamentais da Engenharia de Software	9
Figura 3. Abordagens do Modelo CMMI.....	12
Figura 4. Utilização da Norma ISO 15504.....	19
Figura 5. Processos da Norma ISO/IEC 14598-1.....	24
Figura 6. Atividades do Gerente de Projetos.....	36
Figura 7. Atividades do Coletor de Informações	37
Figura 8. Fluxo de Atividades do Processo.....	39
Figura 9. Relação entre a Qualidade dos Artefatos, dos Casos de Uso e do Projeto.....	45
Figura 10. Inspector Panel: Tela de Inspeção.....	59
Figura 11. Arquitetura do Inspector Panel	60
Figura 12. Qualidade dos UC's <i>versus</i> Avaliações.....	83
Figura 13. Qualidade do Projeto <i>versus</i> Avaliações.....	84
Figura 14. Exemplo do Progresso da Qualidade do Projeto.....	137
Figura 15. Exemplo do Progresso da Qualidade de Caso de Uso	138

Lista de Tabelas

Tabela 1 – Características da Qualidade Externa e Interna.....	21
Tabela 2 – Características da Qualidade em Uso.....	22
Tabela 3 – Avaliação: Marco Concepção.....	51
Tabela 4 – Avaliação: Marco Elaboração.....	52
Tabela 5 – Avaliação: Marco Construção.....	53
Tabela 6 – Primeira Avaliação: Qualidade dos Casos de Uso por Iterações.....	70
Tabela 7 – Primeira Avaliação: Qualidade dos Casos de Uso por Fase.....	70
Tabela 8 – Segunda Avaliação: Qualidade dos Casos de Uso por Iterações.....	74
Tabela 9 – Segunda Avaliação: Qualidade dos Casos de Uso por Fases.....	75
Tabela 10 – Terceira Avaliação: Qualidade dos Casos de Uso por Iterações.....	79
Tabela 11 – Terceira Avaliação: Qualidade dos Casos de Uso por Fase.....	79
Tabela 12 – Comparação entre as Abordagens.....	90
Tabela 13 – Responsáveis pela Coletas de Dados.....	115
Tabela 14 – Identificação dos Casos de Uso.....	129
Tabela 15 – Progresso da Qualidade.....	135
Tabela 16 – Problemas Identificados.....	138

Lista de Abreviações

ABNT	Associação Brasileira de Normas Técnicas
CASE	<i>Computer-Aided Software Engineering</i>
CMM	<i>Capability Maturity Model</i>
CMMI	<i>Capability Maturity Model Integration</i>
CPD	<i>Copy/ Paste Detector</i>
EPF	<i>Eclipse Process Framework</i>
GQA	Garantia da Qualidade
GUI	<i>Graphical User Interface</i>
ISO	<i>International Organization for Standardization</i>
MPS.Br	Melhoria de Processo do Software Brasileiro
PA	<i>Process Area</i>
PATT	<i>Project Assessment and Tracking Tool</i>
PMD	<i>Programming Meticulous coDe</i>
PPQA	<i>Process and Product Quality Assurance</i>
RUP	<i>Rational Unified Process</i>
SEI	<i>Software Engineering Institute</i>
UC	<i>Use Case</i>
UML	<i>Unified Modeling Language</i>
XML	<i>eXtensible Markup Language</i>

1. INTRODUÇÃO

O ser, o ter e o fazer são como triângulo, no qual cada lado serve de apoio para os demais. Não há conflito entre eles.

—SHAKTI GAWAIN

Neste capítulo introdutório é apresentada a importância da qualidade dos produtos de software nos dias atuais, enfatizando o contexto e as motivações para realização deste trabalho de mestrado. A seguir, os objetivos a serem alcançados na conclusão do trabalho são definidos. Além disso, apresenta-se a metodologia usada no desenvolvimento desta dissertação de mestrado. Por fim, a organização do texto do trabalho é descrita.

1.1. Contextualização

Com a evidência da globalização e a internacionalização do mercado, aliado à popularização dos computadores pessoais se tornou inevitável a necessidade do uso de softwares no dia a dia. A importância do software de computadores tem passado por significativas mudanças nos últimos 50 anos (Pressman 2006). No início, com os softwares de prateleiras produzidos em larga escala, de maneira uniforme, e colocados no mercado à disposição de pessoas interessadas. Posteriormente, os softwares vieram embutidos em uma gama enorme de produtos de uso considerado indispensável nos dias atuais, a exemplo de forno microondas, geladeiras, máquinas de lavar, carros, celulares e relógios. Para atender o mercado, em crescente expansão, novas empresas de tecnologia surgem a cada dia, aumentando assim, a competitividade.

O aumento da competitividade entre as empresas de desenvolvimento de software tem feito com que elas busquem cada vez mais produzir software com a qualidade desejada (Vicente 2008). Caso o produto de software produzido não tenha o mínimo de qualidade aceitável e não atenda aos requisitos especificados, o projeto está fadado ao fracasso. Os projetos para desenvolvimento de software costumam ter inúmeros problemas relacionados à gerência, tais como custos excessivos, atrasos no cronograma e, conseqüentemente, o não atendimento às necessidades dos clientes, comprometendo, dessa forma, a qualidade do produto final (Kotonya 1998).

Qualidade é um objetivo a ser buscado nas organizações, pois desenvolver ou selecionar produtos de software de alta qualidade é de primordial importância (ABNT 2003). O termo qualidade pode ter significados diferentes para pessoas diferentes, já que se trata de um termo subjetivo. Baker (2007) afirma que a beleza está nos olhos de quem vê, assim é a qualidade.

A qualidade do produto, para algumas pessoas, pode estar resumida ao bom funcionamento do produto. Para outras pessoas, a qualidade do produto está associada à qualidade da matéria prima utilizada na produção do produto, indo além do bom funcionamento do produto. Sendo assim, cada pessoa deve decidir o que entende por qualidade. Nas organizações não é diferente, sendo necessário definir um conceito do termo qualidade, ou seja, todos precisam ter um entendimento comum do termo qualidade. Este conceito será abordado, em detalhes, no próximo capítulo.

O crescimento da importância da qualidade em software nas organizações se deu pela alta gerência em obter visibilidade nos seguintes aspectos: (i) quanto à qualidade dos processos executados dentro da organização; e (ii) quanto à qualidade dos produtos de software que são entregues aos clientes, além de apoiar a tomada de decisões estratégicas de negócio (Montoni 2006). A qualidade de um produto deve ser inserida na execução do processo de construção e não após a sua finalização. Ou seja, a qualidade deve estar presente desde o início do ciclo de vida do software.

1.2. Motivação

O sucesso é o principal objetivo a ser alcançado por pessoas e organizações. Para tal, se faz necessário planejar e executar várias etapas que levam ao sucesso. As organizações visam o sucesso em seus projetos de software. A literatura enfatiza que para obter o sucesso em projetos de software é necessário atender os requisitos, no prazo estabelecido, dentro do custo e de qualidade aceitável (Humphrey, 1990; Kerzner, 2002; PMBOK, 2004).

Monteiro (2005) enfatiza que para garantir o sucesso do projeto de software dentro do prazo e dos custos aceitáveis, além de um bom planejamento do projeto, é necessário medir o esforço, medir o tempo real para realização de cada atividade e garantir a qualidade dos artefatos gerados durante o ciclo de vida do software.

Devido à existência de projetos de software cada vez mais complexos e com muitas pessoas envolvidas, o gerenciamento de projetos tem participação vital para o sucesso do projeto, atuando durante todo o seu ciclo de vida (Vicente 2008). Este trabalho aborda a qualidade do produto de software, pois de que adiantaria entregar os produtos dentro do prazo, dentro do custo, porém sem qualidade. A falta de qualidade no software produz uma sensação de insegurança, de insatisfação, de inutilidade, de tempo e de dinheiro perdido. Pode ser resumido na seguinte palavra: fracasso.

A qualidade é um fator crítico de sucesso para indústria de software (Softex 2009). Garantir a qualidade do produto de software pode resultar nas seguintes vantagens: redução dos custos com o retrabalho, conseqüentemente, aumento da produtividade e dos lucros da empresa, conquista de mercado – novos negócios – e, principalmente, satisfação do cliente ou usuário final (Deming 1990). Estas vantagens são importantes para subsistência da organização. A competitividade e lealdade de clientes a longo prazo são cada vez mais definidas por qualidade objetivamente experimentável (Mangold 2007). Desta forma, a qualidade deve ser gerenciada de modo apropriado em cada estágio do ciclo de vida do software (ABNT 2003).

1.3. Objetivos do Trabalho

Nesta seção será descrito o objetivo geral e os objetivos específicos dessa dissertação, o qual direcionou todos os esforços e as pesquisas realizadas.

1.3.1. Objetivo Geral

O objetivo principal deste trabalho é definir um processo para avaliar e acompanhar o progresso da qualidade do produto em projetos de software durante todo o ciclo de vida do software, baseado em normas e modelos de qualidade nacional e internacional.

1.3.2. Objetivos Específicos

Os objetivos específicos a que se propõe este trabalho são os seguintes:

- Estabelecer métricas de progresso para avaliação de qualidade dos casos de usos em projetos de software que utilizam a metodologia *Rational Unified Process* (Kruchten 2003) de desenvolvimento. Sendo assim, realizar o

acompanhamento da qualidade em projetos de softwares iterativos, incrementais, orientados a objetos e baseados em casos de usos.

- Estruturar o *workflow*¹ do processo de avaliação de progresso da qualidade de software utilizando o *Eclipse Process Framework* (EPF). O EPF é uma ferramenta que permite a criação e definição de processos customizados para organizações.
- Desenvolver uma ferramenta para oferecer suporte às diversas atividades do processo apresentando indicadores, gráficos e tabelas, sobre o andamento de progresso da qualidade do produto.
- Realizar um estudo de caso com o objetivo de verificar se o processo definido atende as reais necessidades da indústria de software e capturar *feedback's* dos especialistas da área da qualidade de software.

1.4. Metodologia

A natureza deste trabalho está caracterizada como pesquisa aplicada, onde objetiva gerar conhecimento para aplicação prática dirigida à solução. Quanto à forma de abordagem, o presente trabalho é classificado como pesquisa qualitativa, envolvendo um estudo de caso de caráter exploratório. O desenvolvimento desta dissertação dividiu-se em quatro momentos (fases), conforme Figura 1.

¹ Workflow – conjunto coordenado de atividades, seqüências ou paralelas, interligadas com o objetivo de alcançar uma meta comum.

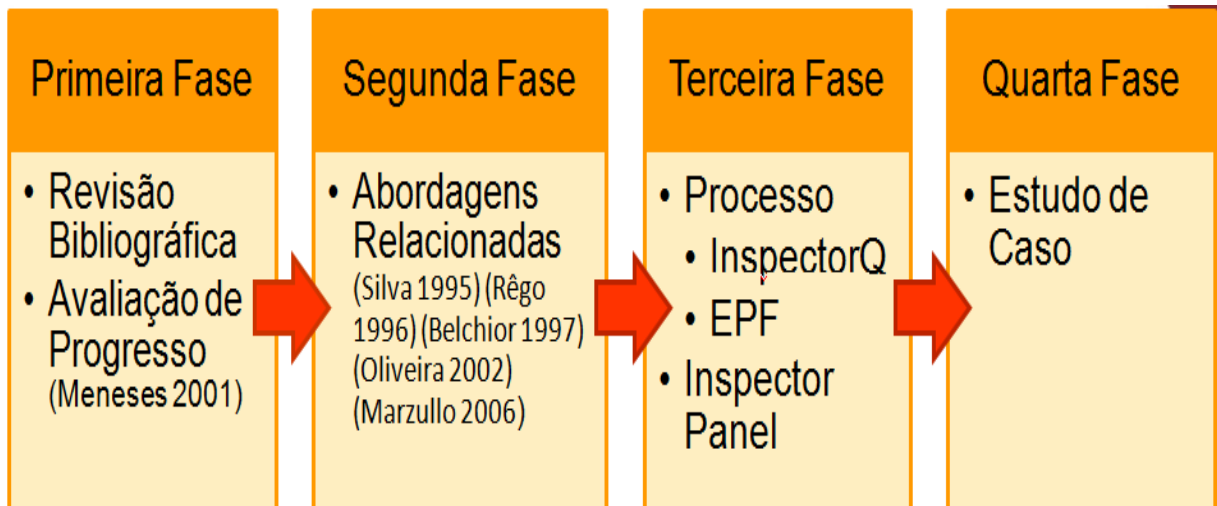


Figura 1. Metodologia do Trabalho

No primeiro momento, foi feita uma revisão bibliográfica em diversas fontes da literatura, visando adquirir embasamento para o desenvolvimento deste trabalho. As fontes pesquisadas foram livros, teses, dissertações, artigos, normas e modelos (nacionais e internacionais). Posteriormente, um estudo detalhado do processo de avaliação de progresso para projeto de software, que objetivou fornecer ao gerente do projeto um conjunto de diretrizes e atividades, que permitirão identificar problemas imediatos e analisar a possibilidade de problemas futuros, o Inspector (Meneses 2001).

No segundo momento, foi feita uma análise de outras abordagens relacionadas ao tema proposto, onde foram identificados os pontos chaves, as lacunas a serem abordadas e melhoradas, bem como a delimitação do escopo do trabalho. Esta análise é abordada na seção de trabalhos relacionados.

O terceiro momento deu-se pela definição de um conjunto de atividades, referentes ao processo de avaliação de progresso da qualidade, que servirão de diretrizes para aplicação do processo em projetos de software em empresas desenvolvedoras de software. O processo de avaliação de progresso da qualidade foi baseado nas etapas da norma ISO/IEC 14598 – Guias para Avaliação de Produto de Software, que fornece métodos para medir, coletar e avaliar a qualidade de produtos de software (ISO/IEC 1996). A partir daí, foi estruturado o *workflow* do processo utilizando o *eclipse process framework* e, posteriormente, construindo uma

ferramenta para apoiar as atividades do processo, calcular as métricas de qualidade estabelecidas e geração de indicadores, gráficos e tabelas, de progresso.

E, finalmente, o quarto momento refere-se à aplicação do processo de avaliação de progresso da qualidade em um projeto de desenvolvimento de software em uma secretaria do governo do estado de Pernambuco.

1.5. Estruturação da Dissertação

Além deste capítulo introdutório, que aborda o contexto, a motivação, os objetivos e a metodologia adotada no desenvolvimento dessa dissertação, este trabalho está organizado da seguinte forma:

- Capítulo 2 – Referencial Teórico: será apresentado o estado da arte na qualidade de produtos e processos de software, enfatizando as principais normas e modelos existentes na literatura, bem com os conceitos usados neste trabalho.
- Capítulo 3 – O Processo: será proposto um processo para avaliação de progresso da qualidade em projetos de software, que utilizam como metodologia de desenvolvimento o RUP (Kruchten 2003) ou uma instância do mesmo. Sendo assim, um conjunto de atividades e métricas para avaliar a qualidade do produto durante as fases e iterações do projeto.
- Capítulo 4 – Estudo de Caso: será apresentado o estudo de caso realizado e resultados alcançados utilizando a proposta deste trabalho, focando nos ganhos obtidos e nos problemas encontrados. O objetivo é obter uma validação para a proposta e para a ferramenta desenvolvida.
- Capítulo 5 – Conclusões: são apresentadas as conclusões do trabalho, propondo as extensões e melhorias para trabalhos futuros.

Ao final são disponibilizados as Referências Bibliográficas, Anexos e Apêndices:

- Apêndice A – Critérios de Avaliação da Qualidade dos Artefatos: apresenta-se o modelo que estabelece os critérios de avaliação da qualidade.
- Apêndice B – Checklist de Avaliação da Qualidade dos Artefatos: documento que contém informações sobre os itens que devem ser observados, durante a avaliação.
- Anexo A – Plano de Avaliação de Progresso: apresenta o modelo para o planejamento da avaliação da qualidade a ser realizada.
- Anexo B – Formulário para Coleta de Informação de Progresso sobre o Caso de Uso: documento que registra as informações obtidas durante a avaliação da qualidade para cada caso de uso.
- Anexo C – Formulário para Coleta de Informação de Progresso sobre o Projeto: documento que registra as informações obtidas durante a avaliação da qualidade do projeto.
- Anexo D – Relatório de Avaliação do Progresso de Qualidade: apresenta o modelo que contém as informações resultantes da análise dos dados obtidos na coleta sobre a qualidade do projeto.

2. REFERENCIAL TEÓRICO

*O conhecimento é o processo de acumular dados;
a sabedoria reside na sua simplificação.*
—MARTIN H. FISCHER

Neste capítulo será apresentado o estado da arte da qualidade de produtos e processos de software, enfatizando as principais normas e modelos existentes na literatura, bem como apresentando os principais conceitos. Inicialmente, uma breve introdução do conceito de qualidade de software para que se tenha um entendimento comum e mais preciso do assunto.

A seguir, uma explanação de normas ISO e dos modelos, CMMI e MPS.Br, para fundamentar o trabalho. Além disso, a garantia da qualidade em projetos de software é apresentada em diferentes abordagens. O papel e a importância do analista da qualidade são descritos em seguida. Por fim, uma explanação do processo de avaliação de progresso em projetos de software orientados a objetos, o Inspector (Meneses 2001), é feita.

2.1. Qualidade de Software

A qualidade de software é uma das áreas de conhecimento da Engenharia de Software que tem como objetivo garantir a qualidade do produto de software através da definição e normatização de processos de desenvolvimento. A Engenharia de Software surgiu durante a década de 1970, com o objetivo de contornar a crise de software e dar um tratamento mais sistemático e controlado da engenharia ao processo de desenvolvimento de software complexos.

Para Borges (2006) a Engenharia de Software é uma área de conhecimento voltada para a especificação, desenvolvimento e manutenção de sistemas de software aplicando tecnologias e práticas de Ciência da Computação, Gerência de Projetos, Gerência de Configuração e outras disciplinas, objetivando organização, produtividade e a qualidade.

Segundo Pressman (2006), a Engenharia de Software abrange métodos, ferramentas e procedimentos no processo de desenvolvimento de software. A Figura 2 apresenta a relação entre os métodos, ferramentas e procedimentos da Engenharia de Software, onde os métodos

detalham como fazer para se construir o software, as ferramentas proporcionam apoio automatizado ou semi-automatizado aos métodos, e os procedimentos constituem o elo que mantém juntos os métodos e as suas ferramentas. Métodos, ferramentas e procedimento possibilitam um processo de desenvolvimento de software claro, eficiente, visando garantir ao desenvolvedor e seus clientes, a produção de um software de qualidade.



Figura 2. Elementos Fundamentais da Engenharia de Software

Obter qualidade nos produtos e nos processos de software não é uma tarefa trivial. Diversos fatores podem dificultar o alcance dos objetivos de qualidade na organização. A falta de qualidade no software pode ser oriunda da falta de documentação do sistema ou documentação pobre, *bugs* na aplicação, entre outros. A descoberta e o reparo de *bugs* no software após o desenvolvimento custa caro (Machado 2001). Por este motivo, é importante avaliar a qualidade do sistema com relação às funcionalidades, confiabilidade, desempenho, usabilidade, entre outros fatores.

Diversas definições de gestão de projetos (Kerzner, 2002; Prado, 2003; PMBOK, 2004) enfatizam a qualidade como um dos aspectos importantes e que deve estar presente durante todo o ciclo de vida de software. O Guia PMBOK trata a qualidade como uma das áreas de conhecimento em gerenciamento de projetos. Esta área, denominada de gerenciamento da qualidade do projeto, descreve todos os processos envolvidos no planejamento, monitoramento, controle e na garantia de que o projeto atenderá aos requisitos de qualidade definidos (PMBOK 2008).

De acordo com Mangold (2007), a qualidade está fundamentada em três alicerces:

- Procedimento estruturado, determinando o que deve ser processado e em seqüência, a fim de obter determinado resultado;

- A utilização de acessórios e ferramentas adequadas, para apoiar os procedimentos; e por fim,
- O treinamento necessário para executar o procedimento utilizando as ferramentas e acessórios.

Para melhorar a qualidade do software são utilizadas estratégias de qualidade, que de acordo com Belchior (1997), são selecionadas com base em alguns critérios, a exemplo de:

- Buscar melhorar os processos, já que alguns erros no produto são conseqüências do processo utilizado;
- O esforço para realizar a identificação e correção de defeitos; e,
- A identificação da causa dos erros que produziram defeitos.

A seguir será apresentado o conceito de qualidade de software e algumas normas e modelos internacionais que servem de referência teórica na definição de atividades que visam obter qualidade nos produtos desenvolvidos.

2.1.1. Conceito de Qualidade

A qualidade é um objetivo a ser alcançado na organização (Rocha 1993) e um dos objetivos da Engenharia de Software (Pressman 2006). O termo “qualidade” pode ser descrito de formas diferentes dependendo da percepção das pessoas e do contexto onde se está inserido, já que, a qualidade de software é algo factível, relativo, substancialmente dinâmico e evolutivo, adequando-se ao nível dos objetivos a serem atingidos (Belchior 1997). A qualidade é subjetiva e varia com o local e com a época, além disso, o mesmo produto pode ser considerado com maior e menor qualidade quando avaliado por diferentes pessoas (Oliveira 2007).

Crosby (1979) define a qualidade como a conformidade aos requisitos estabelecidos no projeto. Para Deming (1982) a qualidade é definida pelo cliente, enquanto que Juran (1988) identifica qualidade como sendo adequação ao uso. Cooper (2002) defende a qualidade com sendo o grau em que um objeto – produto, processo ou serviço – satisfaz um determinado conjunto de atributos ou requisitos.

Segundo a ISO 8402, a qualidade é a totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas (ISO/IEC 1994). As características são exemplificadas e descritas pela ABNT (2003).

Para o contexto deste trabalho, o conceito de qualidade de software adotado foi de (Crosby 1979), onde a qualidade é definida pela conformidade aos requisitos definidos e estabelecidos no projeto. A conformidade é definida pelo atendimento a requisitos que foram especificados e a não-conformidade é definida pelo não atendimento a um requisito especificado.

Para ISO/IEC (1994), o termo “qualidade” não deve ser usado isoladamente para exprimir um grau de excelência no sentido comparativo e nem em avaliações técnicas no sentido quantitativo. Sendo assim, os termos a serem usados no sentido comparativo e no sentido quantitativo, respectivamente, são: qualidade relativa e nível de qualidade. O termo “nível de qualidade” será bastante utilizado neste trabalho, pois são apresentados resultados de avaliações quantitativas.

2.2. Normas e Modelos de Qualidade

Para empresas desenvolvedoras de software, alcançar competitividade pela qualidade implica tanto na (i) melhoria da qualidade dos produtos e de serviços correlatos, como dos (ii) processos de produção e de distribuição de software (Softex 2009).

Diversos estudos e discussões sobre a importância da qualidade sobre o processo de software levaram a indústria a se aliar à academia para o desenvolvimento de padrões e modelos, que pudessem trazer a qualidade de software desejada para as necessidades do mercado de software (Nojoza 2005). A seguir, as subseções apresentam as principais normas e modelos.

2.2.1. CMMI

O modelo *Capability Maturity Model Integration* (CMMI) foi projetado para guiar as empresas desenvolvedoras de software no processo de seleção das estratégias de melhoria, com o objetivo de determinar a maturidade atual do processo e identificar as questões críticas para a qualidade e melhoria do processo de desenvolvimento (Chrissis 2003). O CMMI é um

modelo descritivo e normativo que descreve os atributos essenciais que são esperados para caracterizar o nível de maturidade de uma determinada organização (Paulk 1993).

O modelo foi criado pelo *Software Engineering Institute* (SEI) como integração e evolução dos modelos *Capability Maturity Model for Software* (SW-CMM); *System Engineering Capability Model* (SE-CMM), e *Integrated Product Development CMM* (IPD-CMM). O SEI é uma entidade mantida pelo governo norte-americano e patrocinada pelo departamento de defesa, com o objetivo de auxiliar as organizações no processo de desenvolvimento de software.

A arquitetura do CMMI, apresentada na Figura 3, é constituída por um conjunto de Áreas de Processo – *Process Area* (PA) – dispostas em duas abordagens: (i) modelo por estágio; e, (ii) modelo contínuo. Cada área de processo é baseada na sua finalidade, na relação com outras áreas de processo, nas metas genéricas e específicas, nas referências introdutórias, nos produtos das atividades, nas práticas e sub-práticas genéricas e específicas (Moreira 2004).

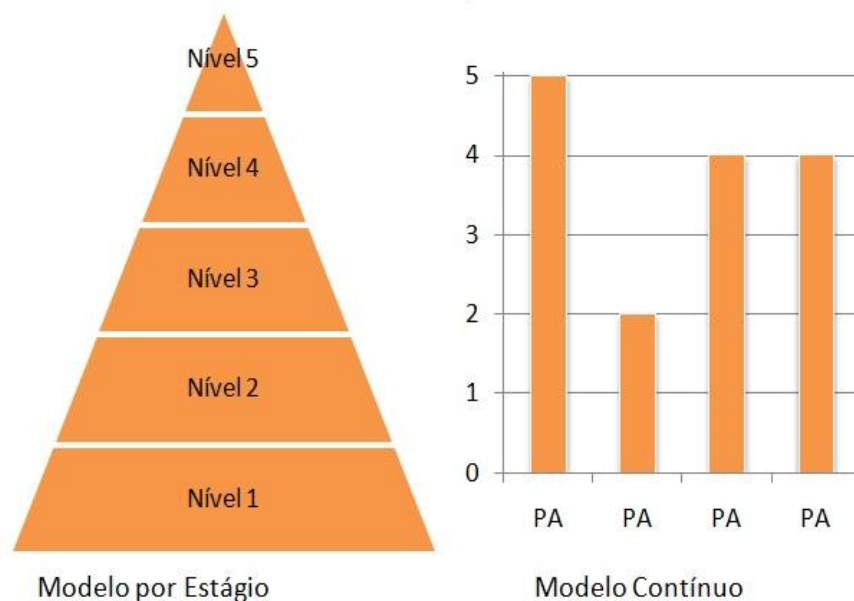


Figura 3. Abordagens do Modelo CMMI

Na abordagem do modelo por estágio, a organização precisa atingir, alcançar, um conjunto de objetivos definidos para todas as áreas de processo de um determinado nível de maturidade. Já a abordagem do modelo contínuo é uma abordagem mais flexível, permitindo a organização escolher e definir as áreas de processo que serão trabalhadas.

O CMMI é dividido em cinco níveis de maturidade e cada nível possui uma caracterização bem específica, descrita a seguir:

- Nível 1 – Inicial: Caracterizado como *ad hoc* e caótico. Neste nível os projetos podem ter seus compromissos não atingidos no prazo estabelecido, os processos (se existirem) são deixados de lado em períodos de crise e não é possível repetir os sucessos do passado. A organização é desprovida de processos e o desempenho do projeto depende das habilidades, dos conhecimentos e das motivações de cada profissional.
- Nível 2 – Gerenciado: Caracterizado pelo gerenciamento, planejamento e execução de atividades de forma padronizada. Neste nível de maturidade os processos, produtos de trabalho e serviços são gerenciados e controlados na organização, sendo possível, gerenciar um ciclo de desenvolvimento estabelecendo políticas para os projetos. Este nível, também, conhecido como repetível, pois os processos são disciplinados para possibilitar a repetição de sucesso em projetos similares.
- Nível 3 – Definido: Caracterizado por um processo padrão estabelecido e documentado. Neste nível todos os projetos utilizam os mesmos processos de desenvolvimento ou manutenção, com pequenas adaptações a natureza específica do projeto e, principalmente, para obter os melhores resultados. Um grupo de engenheiros de processos é responsável pela organização dos processos e das atividades relacionadas.
- Nível 4 – Gerenciado Quantitativamente: Caracterizado por métricas quantitativas. Neste nível a organização coleta métricas para aprimorar suas atividades em busca de melhores resultados. O processo e o produto de

software são compreendidos e controlados quantitativamente. Espera-se alta qualidade dos produtos de software que são gerados.

- Nível 5 – Otimizado: Caracterizado pela melhoria contínua. Neste último nível, a organização foca na melhoria de seus processos através de *feedback's*, acrescido de novas idéias e suportes tecnológicos. O objetivo é prevenir a ocorrências de defeitos, fortalecendo o processo proativamente.

O CMMI trata a garantia de qualidade como sendo uma área de processo do nível 2 – gerenciado do modelo. Categorizada como processo de suporte, a Garantia de Qualidade de Processo e de Produto – *Process and Product Quality Assurance (PPQA)* – possui como propósito prover visibilidade para a equipe técnica e para o gerenciamento, informações objetivas sobre processos e produtos de trabalho associados (Chrissis 2003).

No CMMI, o escopo da atividade de garantia de qualidade pode abranger as avaliações de processos, produtos e serviços; identificação e documentação de não conformidades; fornece *feedback* aos gerentes e a toda equipe do projeto os resultados das atividade de garantia da qualidade; e as atividades não vinculadas a projetos, como treinamento.

As práticas da área de processo, garantia de qualidade de processo e de produto, visa assegurar que os processos previstos foram implementados, enquanto que as práticas na área de processo de verificação assegurar que os requisitos especificados estão sendo satisfeitos. A objetividade nas avaliações da garantia da qualidade do processo e do produto é crítica para o sucesso do projeto. Os objetivos gerais e específicos da área de processo são descritos a seguir:

- Objetivo Geral 1 – Avaliar Objetivamente Processos e Produtos de Trabalho: Neste sentido a aderência do processo definido e executado na organização, os produtos de trabalho resultantes do processo e os serviços associados às descrições do processo, padrões e procedimentos aplicáveis na organização é avaliado objetivamente. Para este objeto geral, temos os seguintes objetivos específicos:

- Objetivo Específico 1.1 – Avaliar Objetivamente os Processos: Avaliar objetivamente os processos definidos executados contra as descrições de processo, padrões e procedimentos aplicáveis. Sendo assim, é preciso promover um ambiente adequado entre todos da equipe, estabelecendo critérios para a avaliação dos processos baseados nos padrões, identificando as não conformidades encontradas, bem como as lições aprendidas durante a avaliação dos processos.
- Objetivo Específico 1.2 – Avaliar Objetivamente os Produtos de Trabalho e Serviços: Avaliar objetivamente os produtos de trabalho e serviços definidos contra as descrições de processo, padrões e procedimentos aplicáveis. Seleção dos produtos de trabalho e serviços a serem avaliados, estabelece critérios para avaliação dos produtos de trabalhos resultantes das atividades do processo e identificação das não conformidades encontradas durante a avaliação dos produtos de trabalho e serviços.
- Objetivo Geral 2 – Fornecer um Entendimento Objetivo:
 - Objetivo Específico 2.1 – Comunicar e Assegurar a Resolução das Questões de Não Conformidades: Identificadas as não conformidades durante a avaliação de processos e produtos de trabalho, é preciso documentar e comunicar as questões de qualidade para assegurar a resolução de questões de não conformidades com a equipe e os gerentes.
 - Objetivo Específico 2.2 – Estabelecer Registros: Estabelece e mantém os registros das atividades de garantia da qualidade do processo e do produto com detalhes suficientes para que o status e os resultados sejam divulgados e conhecidos por todos da equipe do projeto.

2.2.2. MPS.Br

O modelo, criado em 2003, é coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) e tem o apoio do Ministério da Ciência e Tecnologia (MCT), da Financiadora de Estudos e Projetos (FINEP), do Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) e Banco Interamericano de Desenvolvimento (BID) (Softex 2009).

O MPS.Br é um programa para Melhoria de Processo do Software Brasileiro, que visa aprimorar a qualidade no desenvolvimento de software em empresas brasileiras. O modelo MPS.Br é compatível com os padrões de qualidade aceitos internacionalmente e que tenha como pressuposto o aproveitamento de toda a competência existente nos padrões e modelos de melhoria de processo já disponíveis (ABNT 1997; SEI, 2002; ISO/IEC, 2003).

A ISO/IEC 12207, a ISO/IEC 15504 e o modelo CMMI serviram de base técnica para definição do modelo MPS.Br. A primeira norma, ISO/IEC 12207, estabelece uma arquitetura comum para o ciclo de vida de processos de software com uma terminologia bem definida (Softex 2009). Já a segunda norma, ISO/IEC 15504, trata-se das avaliações de processos de software com o objetivo de melhorar os processos na organização e o modelo CMMI determina a capacidade de processos de uma unidade organizacional. O modelo CMMI foi abordado na Seção 2.2.1 desta dissertação.

O MPS.Br apresenta 7 níveis de maturidade, onde o nível A é o último nível a ser alcançado pela organização e o nível G, representa o menor nível do modelo. Os níveis são descritos como segue:

- Nível A – Em Otimização: Composto pelos processos dos níveis de maturidade anteriores (G ao B). Este nível busca a otimização contínua, atendendo integralmente todos os atributos de processos.
- Nível B – Gerenciado Quantitativamente: Composto pelos processos dos níveis de maturidade anteriores (G ao C). Neste nível são acrescentados novos resultados para atender aos objetivos de gerenciamento quantitativo.

- Nível C – Definido: Composto pelos processos dos níveis de maturidade anteriores (G ao D), mais os processos de desenvolvimento para reutilização, gerência de decisões e de riscos. Neste nível o processo já está bem definido e maduro, sendo possível reutilizar componentes de software.
- Nível D – Largamente Definido: Composto pelos processos de níveis anteriores, acrescido dos processos de desenvolvimento de requisitos, de integração do produto, de projeto e construção do produto, de validação e verificação.
- Nível E – Parcialmente Definido: Composto pelos processos de avaliação e melhoria do processo organizacional, pela definição do processo organizacional, pela gerência de recursos humanos e de reutilização. Este nível tem como foco a padronização dos processos organizacionais e já é possível estabelecer, algumas, boas práticas na organização.
- Nível F – Gerenciado: Composto pelos processos de aquisição, de garantia da qualidade e medição, de gerência de configuração e de portfólio de projetos. Neste nível os artefatos produzidos estão de acordo com os padrões estabelecidos e a gestão do projeto possibilita maior visibilidade nas etapas do projeto.
- Nível G – Parcialmente Gerenciado: Composto pelos processos gerência de projetos e de requisitos. Neste nível a organização é orientada a projetos, onde é preciso redefinir algumas operações e atividades, além de estabelecer objetivos, prazos e escopo.

O MPS.Br trata a qualidade como sendo um processo do nível F – Gerenciado. O processo da Garantia da Qualidade (GQA) tem como objetivo garantir que os produtos de trabalho e os processos estabelecidos estejam em conformidade com os padrões, procedimentos e planos estabelecidos pela organização. Os seguintes resultados são esperados:

- GQA 1 – A aderência dos produtos de trabalho aos padrões, procedimentos e requisitos aplicáveis é avaliada objetivamente, antes dos produtos serem entregues e em marcos predefinidos ao longo do ciclo de vida do projeto;
- GQA 2 – A aderência dos processos executados às descrições de processo, padrões e procedimentos é avaliada objetivamente;
- GQA 3 – Os problemas e as não-conformidades são identificados, registrados e comunicados; e,
- GQA 4 – Ações corretivas para as não-conformidades são estabelecidas e acompanhadas até as suas efetivas conclusões. Quando necessário, o escalamento das ações corretivas para níveis superiores é realizado, de forma a garantir sua solução;

Uma avaliação objetiva tem como objeto avaliar as atividades e produtos de trabalho com base em critérios que minimizem a subjetividade e o viés do avaliador. Um exemplo de avaliação objetiva é uma auditoria de requisitos, padrões ou procedimentos por um grupo de garantia da qualidade independente (SEI 2006).

2.2.3. ISO/ IEC 15504

A ISO, em 1992, realizou um estudo para identificar as necessidades e exigências para uma norma de avaliação de processo. O estudo concluiu que era necessária a elaboração de um padrão que fosse aplicável à melhoria dos processos na organização, sendo importante determinar a capacidade desta organização. No ano seguinte, em 1993, iniciou o projeto *Software Process Improvement and Capability Determination* (SPICE) com o objetivo de produzir um relatório técnico, mais específico que a norma ISO 9001 e mais abrangente do que os modelos já existentes. O SPICE foi aprovado em 1998 e a norma ISO/IEC 15504 foi publicada em 2003.

A norma ISO/IEC 15504 é um *framework* para processos de software, e tem como objetivo a melhoria e a determinação da capacidade destes processos, usando os seguintes seis níveis de capacitação, que classificam os processos como: nível 0 ou incompleto, nível 1 ou realizado, nível 2 ou gerenciado, nível 3 ou estabelecido, nível 4 ou previsível, e nível 5 ou

otimizado (ISO/IEC 2003). A ISO/IEC 15504, assim como o CMM, é uma das referências utilizadas pelo modelo CMMI, levando em conta as áreas de processo e níveis de maturidade estabelecidos (SEI 2002).

A norma apresenta a avaliação em dois contextos. O primeiro contexto, denominado de melhoria contínua, procura entender o estado atual dos processos, identificando oportunidades de melhoria com foco na melhoria do processo. O segundo contexto, denominado de determinação da capacidade, tem como objetivo determinar a adequação dos processos que são executados, geralmente realizados para uma organização interessada em contratar a organização avaliada como fornecedor.

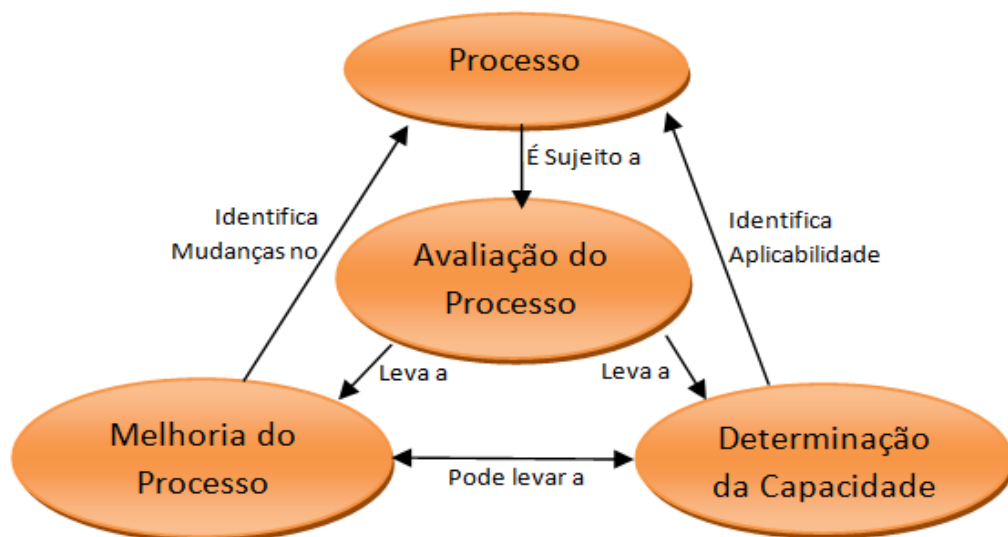


Figura 4. Utilização da Norma ISO 15504

A Figura 4, apresenta a utilização geral da norma onde o processo é o sujeito, foco da avaliação. A avaliação do processo leva a determinação da capacidade da organização e identifica as oportunidades de melhoria no processo avaliado. As melhorias identificadas sugerem pequenas alterações ou reestrutura no processo.

O objetivo da garantia de qualidade, desta norma, é assegurar que os produtos de trabalho e atividades de um processo estão de acordo com os requisitos especificados e satisfaçam aos planos e regras que foram estabelecidas. Os responsáveis pela garantia da

qualidade devem ter autonomia organizacional e autoridade para realizarem as suas tarefas sem interferências dos responsáveis pelo desenvolvimento do software.

O processo de garantia de qualidade está relacionado com os demais processos de suporte estabelecidos na norma, como verificação, validação, revisão conjunta, auditorias e resolução de problemas. Estes processos também são abordados em (ABNT 1997) e suas práticas segundo (ISO/IEC 2003) são:

- Estabelecer uma estratégia de garantia de qualidade;
- Estabelecer padrões de qualidade;
- Definir registros de qualidade;
- Garantir a qualidade das atividades do processo;
- Garantir qualidade dos produtos de trabalho (artefatos);
- Reportar os resultados da qualidade; e
- Tratar os desvios identificados.

Os registros de qualidade são constituídos pelas evidências de que o processo de garantia de qualidade foi realizado (Nojoza 2005).

2.2.4. ISO/ IEC 9126

A primeira versão da norma ISO/IEC 9126 foi disponibilizada em 1991 pela ISO e IEC, sendo revisada em 2001. Esta norma trata da qualidade do produto final e não do processo de desenvolvimento do produto. Em 1996, a ABNT lançou uma versão brasileira, a NBR 13596. Atualmente, a ABNT disponibiliza a NBR ISO/IEC 9126, válida desde julho de 2003. Esta última versão é equivalente à ISO/IEC 9126:2001 e cancela e substitui NBR 13596:1996. A NBR ISO/IEC 9126 – Engenharia de software – Qualidade do produto, consiste em quatro partes:

- Parte 1 – Modelo de qualidade
- Parte 2 – Métricas externas

- Parte 3 – Métricas internas
- Parte 4 – Métricas da qualidade em uso

A primeira parte descreve um modelo de qualidade do produto de software, composto em duas partes: (i) modelo de qualidade para qualidade externa e interna, (ii) modelo de qualidade para qualidade em uso. As demais partes especificam as métricas para avaliar as propriedades de qualidade mencionadas no modelo.

O modelo de qualidade para qualidade externa e interna especifica seis características para qualidade, enquanto que o modelo de qualidade para qualidade em uso especifica quatro características. Cada característica é subdividida em subcaracterísticas.

A qualidade externa é definida pela totalidade das características do produto de software do ponto de vista externo, ou seja, quando o software está sendo executado em um determinado ambiente. Da mesma forma, a qualidade interna é definida pela totalidade das características do produto de software do ponto de vista interno, ou seja, com relação aos requisitos da qualidade interna relativos a código-fonte, entre outros aspectos (ABNT 2003). Já a qualidade em uso, leva em consideração o ponto de vista do usuário, ou seja, são avaliadas quando o software é utilizado pelo usuário, na realização de suas tarefas dentro de um ambiente de trabalho (Bevan 1997).

A qualidade externa e interna categoriza os atributos de qualidade de software em seis características e vinte e sete subcaracterísticas, apresentados conforme a Tabela 1. Todas as características da qualidade externa e interna possuem uma subcaracterística, em comum, a conformidade. A conformidade compreende ao grau de atendimento da característica.

Tabela 1 – Características da Qualidade Externa e Interna

Características	Descrição
Funcionalidade	Capacidade de prover funções que atendam às necessidades explícitas e implícitas dos usuários. As subcaracterísticas são: adequação, acurácia, interoperabilidade, segurança de acesso e conformidade relacionada à funcionalidade.
Confiabilidade	Capacidade de manter um nível, confiável, de desempenho especificado. As subcaracterísticas são: maturidade, tolerância a falhas,

	recuperabilidade e conformidade relacionada à confiabilidade.
Usabilidade	Capacidade de compreensão, aprendizagem e operação por parte do usuário. As subcaracterísticas são: inteligibilidade, apreensibilidade, operacionalidade, atratividade e conformidade relacionada à usabilidade.
Eficiência	Capacidade de apresentar bom desempenho sem desperdício de recursos. As subcaracterísticas são: comportamento em relação ao tempo, utilização de recursos e conformidade relacionada à eficiência.
Manutenibilidade	Capacidade do software de ser facilmente modificado. As subcaracterísticas são: analisabilidade, modificabilidade, estabilidade, testabilidade e conformidade relacionada à manutenibilidade.
Portabilidade	Capacidade de ser transferido de um ambiente para outro. As subcaracterísticas são: adaptabilidade, capacidade para ser instalada, coexistência, capacidade para substituir e conformidade relacionada à portabilidade.

A qualidade em uso mede o grau no qual os usuários podem alcançar seus objetivos, em um ambiente particular, com efetividade, produtividade, segurança e satisfação, ao invés de medir as propriedades do software em si (Simão 2002). A qualidade em uso categoriza os atributos de qualidade de software em quatro características, conforme Tabela 2.

Tabela 2 – Características da Qualidade em Uso

Características	Descrição
Eficácia	Capacidade de permitir que usuários do software atinjam as metas especificadas com acurácia e completitude.
Produtividade	Capacidade de permitir que os usuários empreguem uma quantidade apropriada de recursos em relação à eficácia obtida.
Segurança	Capacidade de apresentar níveis aceitáveis de riscos de danos a pessoas, negócios, software, propriedades ou ao ambiente.
Satisfação	Capacidade de satisfazer aos usuários.

A obtenção de qualidade em uso é dependente da obtenção da qualidade externa, a qual, por sua vez, é dependente da obtenção da necessária qualidade interna (ABNT 2003). Outros contextos podem ser aplicados a ISO/IEC 9126, a exemplo de (Spinelli 2005), que aborda o conceito de visões para avaliação de produtos de software.

2.3. Avaliação de Produto de Software

Em consequência da necessidade de garantir que os produtos de software satisfaçam aos requisitos de qualidade desejados, diversos métodos e processos estão disponíveis para atender esta necessidade. A avaliação dos produtos de software requer planejamento, controle e uso de técnicas de avaliação apropriadas. As medições podem ser realizadas para planejar e controlar a qualidade do produto durante as fases de desenvolvimento, além de avaliar a qualidade do produto final e aprender sobre o processo de software (Rocha 2001).

A série ISO/IEC 14598 apresenta um conjunto de guias para padronizar, planejar e conduzir uma avaliação de produto de software. Não é objetivo desta norma, avaliar o processo de desenvolvimento do software, nem medir custos. A ISO/IEC 14598 define processo de avaliação para (i) organizações que desenvolvem um novo produto de software – desenvolvedores; (ii) organizações que compram um produto de software – compradores; e (iii) organizações que avaliam os produtos de software – avaliadores. A ISO/ IEC 14598 é dividida em seis partes, conforme listado abaixo:

- Parte 1 – Visão Geral
- Parte 2 – Planejamento e Gestão
- Parte 3 – Processo para Desenvolvedores
- Parte 4 – Processo para Aquisição
- Parte 5 – Processo para Avaliação
- Parte 6 – Documentação de Módulo de Avaliação

A ISO/IEC 14598-1 apresenta uma visão genérica do processo de avaliação da qualidade de produto de software podendo ser aplicada a qualquer fase do ciclo de vida. Este guia contém toda a estrutura de funcionamento da avaliação e define os termos técnicos (ISO/IEC 1996). A Norma 14598-1 descreve o processo de avaliação de produtos de software conforme apresentado na Figura 5.

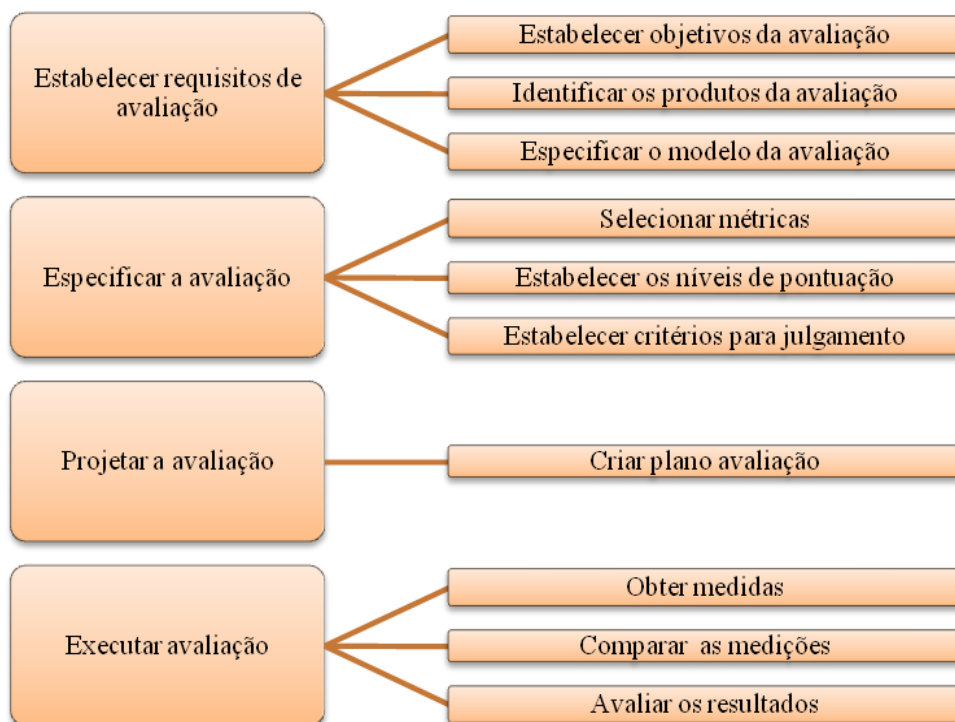


Figura 5. Processos da Norma ISO/IEC 14598-1

O processo é descrito em fases descritas a seguir:

- Estabelecer requisitos de avaliação: Nesta fase (1) estabelece o propósito da avaliação definindo seu objetivo, de forma a garantir que o produto forneça a qualidade necessária; (2) identifica os tipos de produtos a serem avaliados e o estágio atingido pelo produto no ciclo de vida, o que determina quando e qual produto intermediário ou final será avaliado; e, (3) especifica o modelo de qualidade baseada na norma 9126-1.
- Especificar a avaliação: Nesta fase (1) são selecionadas as métricas correlacionadas às características e subcaracterísticas de qualidade; (2) estabelece os níveis de pontuação para as métricas, refletindo em alguma escala a performance particular do sistema a respeito da característica e qualidade em questão; e, (3) estabelece os critérios para julgamento geral do produto, de acordo com a definição de características essenciais, desejáveis.

- **Projetar a avaliação:** Nesta fase, a avaliação envolve a criação de um plano de avaliação, responsável por escrever os métodos de avaliação e um cronograma das atividades do avaliador.
- **Executar avaliação:** Nesta última fase é responsável em (1) obter as medidas em uma pontuação apropriada na escala da métrica utilizada; (2) Comparar o valor medido comparando com critérios estabelecidos predeterminados; e por fim, (3) julgar os resultados do processo de avaliação, onde um conjunto de níveis pontuados é resumido. A qualidade resumida é então comparada com outros aspectos como tempo e custo. Posteriormente, uma decisão gerencial deverá ser tomada com base nos critérios gerenciais e o resultado de aceitação ou rejeição do produto de software deve ser informado.

A ISO/IEC 14598-2 trata do planejamento e da gestão do processo de avaliação do produto de software. Esta parte contém recomendações e orientações que fornecem suporte a avaliação dos produtos em desenvolvimentos, padronização, aquisições, etc. A ISO/IEC 14598-3 seleciona e registra os indicadores que serão medidos e avaliados. Os indicadores são obtidos durante as fases de desenvolvimento e com base nestes indicadores são tomadas decisões gerenciais e estratégicas.

A ISO/IEC 14598-4 estabelece um processo sistemático para avaliar produtos de software comerciais obtidos sob encomenda ou modificações de produtos existentes. Este guia é usada para garantir que os requisitos do produto (desenvolvido ou modificado) sejam atendidos. A ISO/IEC 14598-5 fornece orientações práticas para implementar o processo de avaliação de produto de software quando diversas partes necessitam entender, aceitar e confiar nos resultados da avaliação. A ISO/IEC 14598-6 apresenta a documentação estruturada dos módulos de avaliação.

2.4. Garantia da Qualidade

A garantia da qualidade deverá começar nas fases iniciais de um projeto, para tal é preciso estabelecer planos, processos, padrões e procedimentos que adicionarão valor ao projeto, atendendo aos requisitos do projeto e as políticas organizacionais. A equipe que executa a garantia da qualidade na organização deve participar no estabelecimento dos planos,

processos, padrões e procedimentos, para assegurar que eles se encaixam nas necessidades reais do projeto e que poderão ser utilizados para a execução de avaliações de garantia da qualidade (Chrissis 2003).

Segundo (Pressman 2006), a garantia da qualidade consiste de um conjunto de funções para auditar e documentar a efetividade e completeza das atividades de controle da qualidade. Sua principal meta é fornecer informações necessárias, à gerência, sobre o andamento da qualidade do produto de software. Ou seja, a gerência quer saber se as metas de qualidade estabelecidas para o produto estão sendo alcançadas, de forma satisfatória.

Para (ABNT 1997) o processo de garantia da qualidade é responsável pela definição de atividades visando garantir, objetivamente, que os produtos e processos de software estão em conformidade com os requisitos e planos estabelecidos. As técnicas que podem ser utilizadas para garantir a qualidade do projeto, segundo o processo de ciclo de vida de software são: revisão conjunta, verificação, validação e auditorias.

Para realizar a garantia da qualidade de forma imparcial, a equipe da qualidade deve ter autoridade e autonomia organizacional. Sendo assim, a equipe de qualidade não deve estar subordinada aos responsáveis pelo desenvolvimento do software. Outras formas de tratar a garantia da qualidade podem ser utilizadas, a exemplo da revisão de software. As revisões avaliam o processo de desenvolvimento e seu gerenciamento, além do produto de software (IEEE 1028 1997).

No CMMI, o processo de garantia da qualidade tem um relacionamento grande com a área de processo verificação. A verificação é a confirmação, por exame e fornecimento de evidências objetivas do atendimento aos requisitos especificados (ISO/IEC 1994). As práticas específicas da área de processo de garantia de qualidade garantem que processos que foram planejados sejam, efetivamente, implementados. As práticas específicas da área de processo de verificação garantem que requisitos especificados sejam satisfeitos. Como as duas áreas de processo, garantia da qualidade e verificação, em determinados momentos tratam os mesmos produtos de trabalho, convém que seja minimizada a duplicação de esforços (Villas-Boas 2005).

A garantia da qualidade fornece suporte ao controle da qualidade por meio de evidência e confiança na habilidade do processo empregado em produzir um produto de software que atenda aos requisitos especificados. Cabe à garantia da qualidade verificar se a organização diz o que faz e faz o que diz (Magalhães 2006).

A garantia da qualidade envolve além da definição dos processos na organização, a implementação do processo de garantia da qualidade de software, que seja adequada ao contexto da organização, sempre realizando auditorias de qualidade no produto e no processo (Rocha 2001). A qualidade de processo contribui para melhorar a qualidade do produto e a qualidade do produto, pode indicar melhorias no processo, gerando mais qualidade no processo. Por isso, avaliar e melhorar o processo é um meio de melhorar a qualidade do produto (ABNT 2003).

2.5. Papel do Analista de Qualidade

Os analistas de qualidade têm a missão de auditar independentemente os requisitos do desenvolvimento do software e verificar os produtos de software (Chrissis 2003). As atividades do analista de qualidade, segundo (Unhelkar 2003), têm como objetivo checar atividades e tarefas realizadas pela equipe do projeto, artefatos e modelos UML produzidos, além das conformidades aos padrões de qualidade, identificando técnicas adequadas de qualidade e utilizando *checklists* para garantir a qualidade. Os erros identificados devem ser reportados e avaliações devem ser realizadas após os erros reportados terem sido corrigidos.

De acordo com o SEI as atividades do analista de qualidade são:

- Preparar um plano de garantia da qualidade;
- Participar da descrição do projeto de software;
- Revisar as atividades dos engenheiros de software;
- Documentar e consertar os desvios encontrados;
- Registrar discordâncias e reportar para o gerente; e,
- Gerenciar mudanças e métricas de software.

Marckzak (2003) vai além e acrescenta os seguintes itens:

- Apoiar o uso e compreensão dos processos, procedimentos e templates definidos;
- Apoiar o planejamento do projeto;
- Planejar as atividades de qualidade; e
- Acompanhar a resolução dos desvios encontrados.

Para atuar como analista de qualidade em uma organização é necessário possuir habilidades técnicas e interpessoais, além ter um perfeito conhecimento dos requisitos requeridos pelas normas e modelos existentes e nos processos organizacionais, bem como demonstrar consideração e respeito pelos sentimentos e habilidades dos auditados.

O trabalho de auditar requer confiança, rastreabilidade, cortesia e sensibilidade. Para as organizações, a missão da garantia da qualidade e do analista de qualidade é deixar o ambiente melhor do que encontrou independente do nível de maturidade da organização, a garantia da qualidade é essencial para o sucesso de um programa de melhoria (Magalhães 2006).

2.6. Processo Inspector

O Inspector² é um processo de avaliação de progresso para projeto de software orientado a objetos, que visa fornecer ao gerente do projeto um conjunto de diretrizes e atividades, que permitirão identificar problemas imediatos e analisar a possibilidade de problemas futuros.

A área de concentração deste processo é o progresso no desenvolvimento de um determinado projeto, sendo assim, não focaliza aspectos de qualidade envolvendo o produto final. Apesar disso, o acompanhamento do progresso de um projeto indica que o gerente

² Inspector na web: cin.ufpe.br/~inspector

utiliza métodos e técnicas de gerenciamento, que geralmente resultam em produtos finais entregues na data correta e com qualidade (Meneses 2001).

O processo Inspector busca sistematizar o uso de técnicas e métricas para avaliação do progresso técnico de um projeto de desenvolvimento de software orientado a objetos. Neste contexto, o progresso do projeto é obtido pela comparação dos valores reais com os estimados e quando o status real desvia de forma significativa dos valores esperados ou planejados, devem ser tomadas as ações corretivas apropriadas (Vicente 2008).

O Inspector fornece ao gerente de projeto duas visões de progresso complementares que fornecem suporte para um bom gerenciamento do processo de desenvolvimento. São elas: a visão de desempenho e a visão de funcionalidade.

A primeira visão verifica o desempenho das equipes de desenvolvimento a partir da análise das atividades planejadas para as mesmas, definindo três métricas que mostram a qualidade do planejamento e a produtividade da equipe de desenvolvimento. Esta visão preocupa-se em mostrar quantitativamente problemas de planejamento das atividades das diversas equipes, e o quanto se foi concluído das atividades realmente planejadas. A partir das informações desta visão, pode-se verificar a necessidade de inserir, realocar ou retirar recursos do projeto.

A visão de funcionalidade define uma métrica, dirigida a casos de uso, que verifica o progresso funcional do sistema. Esta visa superar as limitações existentes na visão de desempenho, e tem como objetivo fundamental indicar o aumento percentual (%) da funcionalidade do sistema como um todo, a partir da análise de cada caso de uso do projeto individualmente. De posse das duas visões, o gerente de projeto é capaz de identificar equipes com dificuldades e casos de uso com problemas no desenvolvimento (Meneses 2001).

2.7. Trabalhos Relacionados

Em um estudo de pesquisa-ação, os trabalhos relacionados devem ser úteis à pesquisa no sentido de indicarem como o estudo deve ser conduzido. Para o contexto desta pesquisa, apresenta-se o trabalho de (Silva, 1995; Rêgo, 1996; Belchior, 1997; Oliveira, 2002; Marzullo, 2006).

O trabalho apresentado em (Silva 1995), propõe um ambiente interativo de validação de qualidade de software baseado no teste estrutural de unidade, de integração e de sistema. Desenvolvido na linguagem de programação C, a ferramenta auxilia o desenvolvedor no planejamento da fase de teste, nas tarefas de seleção de caminhos e na sua execução. Silva (1995) afirma que o ambiente é configurável, podendo ser adaptado a diferentes linguagem de programação. Contudo o ambiente pode ser usado, apenas, para linguagens estruturais não sendo aplicável para **atual realidade**.

Por sua vez, o trabalho (Rêgo 1996) é resultado de anos de pesquisas do Centro de Tecnologia da Informação Renato Archer (CTI) e abrange todos os aspectos do produto de software. Baseado na norma 9126 (ABNT 2003), este modelo apresentou pouca flexibilidade e possui foco em software de prateleira.

O modelo baseado na teoria *fuzzy* (Belchior 1997) leva em consideração as características de qualidade e o perfil dos especialistas envolvidos, ambos associados a um grau de importância. O modelo atua como um mecanismo capaz de interpretar os resultados e de sintetizar as informações, através de procedimentos normatizados. (Belchior 1997) enfatiza como os principais objetivos do modelo como segue: (i) Prover suporte ao processo de avaliação da qualidade; (ii) possibilitar a obtenção do julgamento do avaliador, levando em consideração a experiência do mesmo; (iii) representar matematicamente os julgamentos dos avaliadores, apesar da subjetividade dos atributos de qualidade em questão; e, por fim, (iv) obter um resultado quantitativo referente ao nível de qualidade do software.

Posteriormente, uma ferramenta foi desenvolvida para modelar e implementar as etapas definida no modelo de (Belchior 1997) apresentado por (Oliveira 2002). O principal objetivo de (Oliveira 2002) é aumentar a visibilidade na execução dos processos de avaliação da qualidade, devido à grande busca pela qualidade, alinhado a redução de custos e prazo. A utilização deste modelo, sem a ferramenta, requer um conhecimento matemático elevado.

O trabalho proposto por (Marzullo 2006) tem como objetivo principal a concepção de uma abordagem de desenvolvimento orientada por métricas de produto e de qualidade, auxiliando o gerente no controle de seus projetos. Este auxílio dá-se-a através da análise das métricas extraídas de artefatos de software produzidos ao longo do desenvolvimento.

As atividades que fazem parte desta abordagem são: (i) Planejar medições – trata dos aspectos necessários para o devido planejamento do processo de medição, compreendendo as sub-atividades de estabelecimento dos objetivos, especificação das medidas e da coleta de dados, e por fim, definição de mecanismos de armazenamento e de técnicas de análise; (ii) Executar medições – garante a correta execução e utilização dos procedimentos definidos no planejamento, tendo como sub-atividade a execução de coleta de dados; e, (iii) Avaliar medições – esta atividade é responsável por extrair informações capazes de suportar ações corretivas dentro do processo, englobando as sub-atividades de execução de atividades de análise, comunicação ampla das informações obtidas, execução de ações corretivas, finalizando pela avaliação, seleção e priorização das melhores soluções.

Para facilitar a análise e dar suporte à abordagem definida por Marzullo, foi desenvolvida a ferramenta *Project Assessment and Tracking Tool* (PATT) nos laboratórios de engenharia de software e banco de dados da Universidade Federal do Rio de Janeiro (UFRJ). A ferramenta foi desenvolvida utilizando a linguagem de programação Java, apoiada por um banco de dados XML Nativo. O PATT pode ser utilizado no ambiente organizacional, onde o processo de desenvolvimento de software esteja formalizado, baseado em normas e padrões.

Um ponto, positivo, que pode ser observado é o nível de detalhamento que o PATT oferece, a exemplo, quando apresenta os arquivos de fonte, possibilitando assim visualizar o status de progresso dos arquivos (.java, por exemplo). Outro ponto menos positivo do que o descrito anteriormente é a necessidade de tratar os diversos artefatos (documentos, modelos UML, código-fonte), transformando-os em documento XML.

2.8. Considerações Finais

Alcançar competitividade pela qualidade, para as empresas de software, implica tanto na melhoria da qualidade dos produtos de software e serviços correlatos, como dos processos de produção e distribuição de software.

Na literatura encontra-se a qualidade como um dos fatores que levam o projeto ao sucesso, em conjunto com o escopo, com o custo e tempo. Sendo assim, é preciso acompanhar esses fatores durante todo o ciclo de vida do software. O processo Inspector aborda o progresso de projetos de software, com base no incremento das funcionalidades no

software, tratando de certa forma, diretamente, do escopo e indiretamente do tempo. O desempenho das equipes também é monitorado pelo processo Inspector.

O presente capítulo apresentou o conceito da qualidade de software, destacando os principais modelos e normas, nacionais e internacionais. Abordou a garantia da qualidade do projeto de software, destacando o papel do analista de qualidade. Destaca-se, ainda, a importância da qualidade em projetos de software, a relação da influência da qualidade do processo na qualidade do produto. Em seguida apresentou-se o processo Inspector, para avaliar o progresso de projetos de software.

O capítulo seguinte aborda alguns conceitos utilizados pelo RUP (Kruchten 2003), define a qualidade de um caso de uso e como influencia na qualidade do projeto. Posteriormente, o processo de avaliação de progresso da qualidade em projetos de software, é apresentado, detalhando a visão da qualidade estabelecida e suas métricas. Ao final, exemplifica-se a obtenção da visão de qualidade em projetos de software.

3. O PROCESSO: INSPECTOR Q

*Há um tempo em que é preciso abandonar as roupas usadas,
que já tem a forma do nosso corpo, e esquecer os nossos caminhos,
que nos levam sempre aos mesmos lugares.
É o tempo da travessia: e, se não ousarmos fazê-la, teremos ficado,
para sempre, à margem de nós mesmos.
—FERNANDO PESSOA*

O presente capítulo apresenta o processo para avaliar o progresso da qualidade em projetos de software orientados a objetos. O processo denominado de Inspector Q define um conjunto de atividades para o processo de avaliação, onde também foram identificados os papéis responsáveis por cada uma das atividades, os artefatos produzidos para documentar a avaliação e estabeleceu as métricas para o acompanhamento do progresso da qualidade do projeto.

O Inspector Q pode ser aplicado em projetos de pequeno, médio e grande porte, onde é necessário acompanhar a evolução do progresso da qualidade do software em desenvolvimento.

3.1. Conceitos Básicos

Os conceitos adotados no processo de avaliação de progresso da qualidade são idênticos à terminologia utilizada pelo RUP (Kruchten 2003). O RUP é um processo de engenharia de software definido e bem estruturado para desenvolvimento de software. Fundamentado nos princípios de iteração, centrado na arquitetura e guiado por casos de uso, este processo define claramente atividades, artefatos e responsabilidades. Sendo assim, utiliza-se a mesma notação para facilitar o entendimento e permitir a integração entre os processos nas organizações. As subseções apresentam alguns principais conceitos.

3.1.1. Fluxo de Atividades

O fluxo de atividades serve para nortear a avaliação de progresso da qualidade definindo uma seqüência ordenada das atividades a serem executadas. Cada atividade corresponde a uma unidade de trabalho tangível e possui um responsável definido, ou seja, a

atividade é executada por um papel. A atividade é descrita em vários passos de execução e possuem artefatos informados e resultantes ou artefatos de entrada e saída (Kruchten 2003).

3.1.2. Atividades

A atividade é uma unidade de trabalho que uma pessoa executa quando está exercendo um determinado papel e produz um resultado importante para o contexto do projeto. O conceito de papel é definido na Seção 3.1.3 e o como resultado importante são os artefatos de saída gerados pela atividade. Pode-se considerar como resultado importante cada artefato produzido pela atividade.

3.1.3. Papéis

Os papéis definem o comportamento e as responsabilidades de uma determinada pessoa ou a um grupo de pessoas trabalhando na mesma equipe (Kruchten 2003). Os papéis não são indivíduos e nem títulos de trabalho. Um papel é uma definição abstrata de um conjunto de atividades executadas e dos respectivos artefatos (Kruchten 2003). Uma pessoa pode assumir vários papéis, a exemplo de analista de sistema, analista de negócio, engenheiro de software, etc.

3.1.4. Responsável

O responsável corresponde a um papel que pode ser atribuído a uma determinada pessoa ou grupo, requer responsabilidade e ações, tais como realizar atividades e produzir artefatos. O papel é responsável pela manutenção de um conjunto de artefatos e por executar as tarefas definidas no processo. Requer conhecimentos e habilidades específicas na execução das tarefas.

3.1.5. Artefatos

Os artefatos são produtos de trabalho finais ou intermediários produzidos e usados durante o desenvolvimento dos projetos. Os artefatos são usados para capturar, armazenar e transmitir informações sobre o projeto. São de responsabilidade de um único papel e promovem a idéia de que todas as informações no processo devem ser responsabilidade de uma pessoa específica (Kruchten 2003).

3.2. Visão Geral do Processo

O processo apresenta um conjunto de atividades que direcionam para inspeção de artefatos, coletando e analisando as métricas estabelecidas, fornecem ao gerente de projetos informações objetivas e precisas sobre o progresso da qualidade do projeto de software, através da visão de qualidade.

A visão de qualidade define o nível de qualidade das funcionalidades desenvolvidas no software, indicando o acréscimo de qualidade obtido em porcentagem. O objetivo desta visão é calcular o valor do progresso da qualidade de software ($\varphi_{sistema}$), a partir do cálculo do progresso da qualidade para cada caso de uso ($\varphi_{casodeuso}$). Na Seção 3.5 será apresentada, em detalhes, a definição das métricas de progresso do projeto e dos casos de uso, bem como suas propriedades.

A visão de qualidade pode ser apresentada das seguintes formas: tabela de resumo e gráfico de linha. A tabela de resumo, gerada após a avaliação de progresso, fornece informações atuais da qualidade do projeto e de todos os casos de uso avaliados. Na tabela de resumo é possível identificar o progresso da qualidade em cada iteração e fase do projeto de desenvolvimento de software. Outro indicador a ser observado é o gráfico de linha, que indica o progresso da qualidade em relação ao tempo, facilitando a visualização do progresso e permite uma melhor análise de perspectivas futuras.

As subseções seguintes apresentam uma descrição dos papéis envolvidos no processo, os artefatos produzidos, o fluxo de atividades, a visão de qualidade estabelecida e por fim, as considerações finais.

3.2.1. Papéis

O processo identificou a necessidade de apenas dois papéis para a avaliação de progresso da qualidade em projetos de software. No entanto, existe a possibilidade de outros papéis estarem envolvidos durante a avaliação de progresso da qualidade. Os papéis identificados foram o gerente de projetos e o coletor de informações, ambos, detalhados a seguir.

3.2.1.1. Gerente de Projetos

O papel do gerente de projetos é coordenar as interações com os clientes e com os usuários do sistema. O gerente estabelece as metas a serem alcançadas em cada etapa do projeto, para isso alocar os recursos necessários e mantém as equipes centradas nas metas estabelecidas. Este papel também estabelece um conjunto de práticas que garantam a integridade e a qualidade dos artefatos do projeto, com o objetivo de satisfazer as necessidades do cliente.



Figura 6. Atividades do Gerente de Projetos

No processo, Inspector Q, o gerente de projetos é responsável em planejar a avaliação, avaliar os resultados que foram obtidos durante a avaliação e por encontrar soluções para os problemas que foram identificados e documentados, conforme a Figura 6. Certamente é recomendado que o gerente de projetos tenha o apoio e ajuda da equipe de qualidade e de outras equipes.

3.2.1.2. Coletor de Informações

O papel do coletor de informações é responsável pela coleta de informações e pelo cálculo das métricas definidas no processo (Meneses 2001). Ou seja, o coletor de informações desempenha a atividade de coletar e processar dados referentes à qualidade dos casos de uso e do projeto, conforme apresentado na Figura 7. Neste sentido são capturados e inspecionados os artefatos que são produzidos durante o desenvolvimento de software, com o objetivo de identificar a qualidade dos casos de uso, conseqüentemente, a qualidade do projeto.



Figura 7. Atividades do Coletor de Informações

Os artefatos que serão inspecionados são definidos pelo processo de desenvolvimento da empresa desenvolvedora de software. Além do artefato de especificação de casos de uso, outros artefatos deverão ser inspecionados, por exemplo: documento de requisitos, diagramas de UML, plano de projeto, casos de testes, código fonte, etc. A inspeção deve seguir os critérios de inspeção de qualidade definida para cada artefato.

3.2.2. Artefatos Produzidos

O processo define um conjunto de artefatos que servem de apoio para documentar a avaliação da qualidade do projeto. Os templates dos artefatos produzidos são apresentados em detalhes nos apêndices e anexos desta dissertação. Nas próximas subseções, será apresentada uma descrição de cada artefato produzido para o processo de avaliação de progresso da qualidade de software.

3.2.2.1. Plano de Avaliação

Documento que mantém informações sobre o planejamento da avaliação da qualidade a ser realizada. Neste artefato estão presentes as informações sobre o escopo e data da avaliação, os responsáveis e as metas a serem alcançadas na avaliação de qualidade.

O escopo define os casos de uso que serão inspecionados na avaliação, a data da avaliação especifica o dia ou período de dias em que a avaliação ocorrerá. Os responsáveis são as pessoas identificadas que participam da avaliação, inspecionando os artefatos produzidos e verificando as metas alcançadas.

3.2.2.2. Critérios de Avaliação da Qualidade dos Artefatos

Documento que define as etapas necessárias para o desenvolvimento do projeto e estabelece um conjunto de artefatos a serem produzidos em cada etapa. Cada artefato

estabelecido deve ser associado ao artefato *Checklist* de Avaliação da Qualidade dos Artefatos, que serve de apoio durante a avaliação para que se possa obter um indicador do nível de qualidade do artefato em avaliação.

3.2.2.3. Checklist de Avaliação da Qualidade dos Artefatos

Documento que mantém informações sobre os itens que devem ser observados, durante a avaliação, em cada artefato produzido no projeto. Cada *checklist* deve expressar com clareza e objetividade os itens essenciais e necessários que, cada artefato deve cobrir em sua completude (legibilidade, integridade, conformidade, consistência, não ambigüidade, etc), respeitando sempre a natureza específica de cada artefato.

3.2.2.4. Formulário para Coleta de Informações sobre o Caso de Uso

Documento que registra as informações obtidas durante a avaliação da qualidade para cada caso de uso. Este artefato sofre forte influência do artefato Critérios de Avaliação da Qualidade dos Artefatos, pois especificam os artefatos a serem avaliados e em qual fase do projeto o artefato deve ser desenvolvido. Alguns artefatos não, necessariamente, precisam ser produzidos para determinados casos de uso. Sendo assim, deve-se registrar como opcional o artefato.

3.2.2.5. Formulário para Coleta de Informações sobre o Projeto

Documento que registra as informações obtidas durante a avaliação da qualidade do projeto. Este artefato tem informações que identificam o projeto e o responsável pela coleta e preenchimento do artefato.

Identifica os casos de uso do projeto e o nível de qualidade que cada caso de uso se encontra na avaliação. Consta, também, o nível de qualidade do projeto e o incremento percentual obtido na avaliação. A qualidade do projeto deriva diretamente do nível da qualidade em que os casos de uso se encontram na avaliação.

3.2.2.6. Relatório de Avaliação de Progresso de Qualidade

Este documento contém as informações resultantes da análise dos dados obtidos na coleta sobre a qualidade do projeto. Nele são documentados os resultados observados, as tendências para o projeto, além dos problemas que foram identificados durante a avaliação

(Meneses 2001). Composto por gráficos de linhas e tabelas que representam o nível de qualidade dos casos de uso e do projeto.

3.2.3. Fluxo de Atividades

Esta seção apresenta a descrição do fluxo de atividades definida pelo processo, que objetiva prover à organização informações claras de como proceder com a avaliação de progresso da qualidade em projetos de software. A Figura 8 mostra o fluxo de atividades contendo a relação entre as atividades e os papéis envolvidos no processo de avaliação de progresso da qualidade.



Figura 8. Fluxo de Atividades do Processo

As subseções, a seguir, descrevem cada uma das atividades a serem realizadas pela equipe de qualidade.

3.2.3.1. Planejar avaliação

Esta atividade deve ser realizada sempre que se desejar acompanhar a qualidade do projeto em desenvolvimento e tem como objetivo realizar o planejamento da avaliação da qualidade determinando a data da avaliação, os responsáveis e os casos de usos a serem inspecionados.

O artefato produzido por esta atividade é o Plano de Avaliação e tem como responsável o Gerente de Projetos. O gerente de projetos deve estar em sintonia com as demais equipes que participam do projeto de software, a exemplo, a equipe de desenvolvimento e de qualidade, principalmente. A seguir os passos a serem executados durante esta atividade.

- Estabelecer uma data para realizar a avaliação – Consiste em definir uma data ou período no qual os responsáveis executarão a avaliação no projeto,

respeitando o escopo definido no planejamento. Sugere que as avaliações sejam programadas para o final de cada marco do projeto, pois se comparar os resultados obtidos com o planejado.

- Definir o escopo da avaliação – Durante uma avaliação, os responsáveis devem definir claramente qual o escopo da avaliação, ou seja, quais casos de uso serão inspecionados. Ressalta-se que os casos de uso a serem avaliados sejam os casos de uso que estão em desenvolvimento na iteração. Após a definição do escopo, precisa avaliar a complexidade de cada caso de uso.
- Identificar os responsáveis – Os responsáveis recebem tarefas a serem executadas durante a avaliação. É importante que cada pessoa que participar da avaliação tenha suas atribuições bem definidas, bem como deve estar apto a realizar suas tarefas.

3.2.3.2. Coletar e processar dados da qualidade

Esta atividade consiste em recuperar informações relacionadas à qualidade dos casos de uso que fazem parte do projeto e que estão dentro do escopo da avaliação que foi definido no planejamento, bem como apresentar o nível da qualidade do projeto.

O responsável por esta atividade é o Coletor de Informações e produz como saída o artefato Formulário para Coleta de Informações de Progresso sobre o Caso de Uso. O papel de coletor de informações deve ser desempenhado por membros da equipe de qualidade. A seguir os passos a serem executados durante esta atividade.

- Coletar informações do projeto – Consiste no preenchimento de informações básicas, porém relevantes, no artefato Formulário para Coleta de Informações de Progresso sobre o Caso de Uso. Estas informações são relevantes para a identificação do projeto e de casos de usos inspecionados durante a avaliação.
- Identificar os artefatos produzidos pelos casos de uso – Durante todo o ciclo de vida do software são produzidos artefatos, a exemplo de especificação de caso de uso e diagramas de UML. Sendo assim, o Coletor de Informações identifica

os artefatos produzidos para cada caso de uso dentro do conjunto de artefatos existentes no processo de desenvolvimento de software da organização.

- Coletar informações sobre a qualidade de cada caso de uso – O Coletor de Informações deve proceder de forma a verificar a qualidade de cada caso de uso, independentemente. Certamente, este momento demanda tempo e o nível de complexidade é elevado. O Coletor de Informações deve inspecionar o nível de qualidade que se encontra cada artefato produzido. A inspeção dos artefatos deve obedecer aos critérios definidos no artefato Critérios de Avaliação da Qualidade dos Artefatos. Visando facilitar a aplicação dos critérios definidos para avaliação da qualidade dos artefatos foi elaborado o artefato *Checklist* de Avaliação da Qualidade dos Artefatos, que deve ser preenchido para cada artefato inspecionado. No *checklist* deve conter itens de verificação de aspectos de qualidade, tais como, legibilidade, integralidade, consistência, não ambigüidade, entre outros, conforme padrões adotados na organização.
- Calcular a métrica de qualidade – Consiste em calcular o progresso da qualidade obtida durante a avaliação. Este passo pode ser realizado em conjunto com o passo anterior, coleta de informações, ou seja, simultaneamente. Neste sentido o resultado é encontrar o valor de $\varphi_{casodeuso}$.

3.2.3.3. Avaliar os resultados

Posterior ao levantamento das informações necessárias para realização da avaliação é necessário avaliar os resultados que foram obtidos na avaliação. O responsável por esta atividade é o Gerente de Projeto e tem como objetivo verificar e analisar todos os dados levantados pelo Coletor de Informações. A seguir os passos a serem executados durante a atividade.

- Verificar dados levantados – Consiste em analisar os dados levantados pelo Coletor de Informações estão representados de forma correta e clara, observando a veracidade das informações.

- Analisar as métricas de qualidade de cada caso de uso – Corresponde à análise das métricas de qualidade definidas.
- O valor da métrica $\varphi_{casodeuso}$ indica o nível de qualidade obtida para cada caso de uso inspecionado e pode obter o valor máximo igual a 1, ou seja, o nível de qualidade do caso de uso é 100%.
- Calcular a qualidade do projeto – De posse dos valores das métricas de qualidade para cada caso de uso, o Gerente de Projetos possui informações suficientes para calcular o valor $\varphi_{sistema}$ da métrica para todo o sistema e para cada etapa do seu desenvolvimento. Neste momento, o gerente deve utilizar o artefato Formulário para Coleta de Informação de Progresso sobre o Projeto para documentar as informações sobre o projeto e escopo da inspeção, ou seja, os casos de uso do projeto, além da qualidade dos casos de uso e a qualidade do projeto.

3.2.3.4. Solucionar os problemas

O propósito desta atividade é buscar soluções que visam corrigir os problemas encontrados durante a avaliação da qualidade do projeto. Cada problema identificado deve ser tratado adequadamente buscando a melhor solução a fim de eliminar os problemas.

O Gerente de Projetos é o responsável pela coordenação desta atividade. Devem participar da atividade os representantes de todas as equipes envolvidas, diretamente ou indiretamente, a exemplo da equipe de qualidade do projeto, que estabelecerá diretrizes para garantir a qualidade dos casos de uso, além de oferecer apoio às demais equipes quando necessário. A equipe de análise e de implementação serão responsáveis em propor soluções para os desvios encontrados. A seguir os passos a serem executados durante a atividade.

- Identificar o problema – Primeiramente se faz necessário identificar o problema a ser solucionado. Posterior a identificação é preciso expor o problema para que todos os *stakeholders* presentes tenham a ciência do problema que está sendo tratado, e logo em seguida entender claramente o problema.

- Escolher a solução – Após a compreensão do problema, o Gerente de Projetos deve levantar as possíveis soluções juntamente com todos os *stakeholders* envolvidos. Devem-se registrar as soluções sugeridas e qual a solução a ser adotada justificando sua adoção, bem como definir os responsáveis pela aplicação da solução junto ao problema. O Gerente de Projetos deve levar em consideração as atividades já demandadas para cada equipe do projeto, de modo a evitar uma sobrecarga de atividades, conseqüentemente, atrasos no cronograma.
- Executar e acompanhar a solução – O Gerente de Projetos deve revisar o planejamento das atividades do projeto (planejamento de iterações e do projeto), para que reflita a realidade do projeto atual. Paralelamente, os responsáveis em aplicar as soluções definidas para cada problema, devem executar suas atividades respeitando o planejamento das atividades estabelecidas pelo gerente. Certamente, o Gerente de Projetos deverá acompanhar o andamento dessas atividades, juntamente com as demais atividades.

3.3. Definição de Complexidade

Definir um nível de complexidade de uma determinada atividade não é uma tarefa fácil. A complexidade é um substantivo que exprime o grau de dificuldade em realizar ou fazer algo ou alguma coisa. Neste trabalho de dissertação, a complexidade está associada ao grau de dificuldade na realização das atividades, na elaboração dos artefatos do projeto e na natureza do projeto ou tipo de projeto. Projetos de maior complexidade exigem melhores controles e devem ser acompanhados com maior frequência, diferentemente de projeto de baixa complexidade.

Diversos indicadores podem influenciar a definição da complexidade de um projeto de software, a exemplos de:

- Tipo de projeto – manutenção dos casos de uso existentes, inclusão de novos casos de uso ou a construção de um novo sistema;

- Avaliar o impacto do projeto de software nos processos de negócio da organização;
- Quantidade de *stakeholders* envolvidos no projeto ou afetados pelo projeto;
- Quantidade de *stakeholders* envolvidos na definição das regras de negócio;
- Obtenção de dados e captura de informações relevantes do negócio da organização.

Todos estes indicadores devem ser levados em consideração para determinar complexidade do projeto e de seus casos de uso.

A complexidade dos casos de uso tem o seu uso nas métricas de progresso de qualidade, onde valor da complexidade vai ser associado ao peso ($p_{\text{CasodeUso}}$) que cada caso de uso tem no projeto. Da mesma forma, deve-se determinar um peso (p_{artefato}) para cada artefato produzido.

3.4. Qualidade de um Caso de Uso

Um caso de uso provê uma funcionalidade para o usuário do software e sua descrição deve ser feita com base no domínio do problema, não considerando aspectos computacionais. Cada caso de uso deve estar associado a determinado ator. Este ator pode ser um usuário ou um grupo de usuários que utiliza o software.

Definir a qualidade de um caso de uso compreende em definir a qualidade de cada artefato usado na descrição e realização do caso de uso. Casos de uso diferentes podem ser compostos pelos mesmos artefatos. Os artefatos (modelos, diagramas) produzidos, visam registrar e documentar o que cada caso de uso faz no software. Sendo assim, o nível de qualidade de um caso de uso está diretamente relacionado ao nível de qualidade dos artefatos produzidos. Da mesma forma, o nível de qualidade do projeto de software está diretamente relacionado ao nível de qualidade dos casos de uso que compõem o projeto.

A Figura 9 traz a relação entre a qualidade dos artefatos, dos casos de uso e do projeto, onde a qualidade dos artefatos contribui para melhorar a qualidade dos casos de uso e a

qualidade dos casos de uso contribui para melhorar a qualidade do projeto. Sendo assim, a qualidade do projeto depende da qualidade dos casos de uso e a qualidade dos casos de uso depende da qualidade dos artefatos.

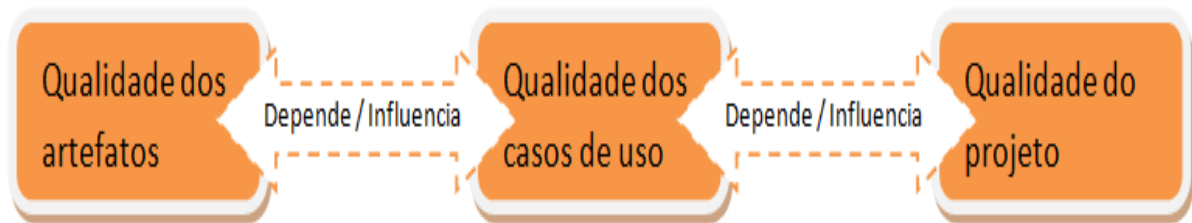


Figura 9. Relação entre a Qualidade dos Artefatos, dos Casos de Uso e do Projeto

A qualidade do artefato é definida através de inspeções realizadas pela equipe de qualidade da organização. As inspeções são um dos quatros componentes básicos apresentados por (Wheeler 1998) como um programa efetivo para garantir a qualidade em projetos de software. Segundo (IEEE 1028 1997), as inspeções são revisões formais realizadas nos artefatos, incluído código fonte. Suas atividades e papéis são bem definidos para realizar a avaliação de qualidade dos artefatos, geralmente, utilizando *checklists*.

3.5. Métrica de Progresso da Qualidade

O progresso corresponde a um indicador que fornece informações precisas sobre o andamento das atividades do projeto. Suponha que um colaborador de uma fábrica seja responsável em produzir uma determinada quantidade de produtos. Os produtos são iguais, tendo as mesmas características. O colaborador tem que produzir durante sua jornada de trabalho dez produtos iguais. Ao final do primeiro turno, o colaborador observou que tinha produzido quatro produtos, então, o progresso das suas atividades foi de quarenta por cento. Ciente do seu progresso, o colaborador, precisaria exercer suas atividades com maior rapidez e sem comprometer a qualidade do produto no segundo turno.

O progresso pode, também, ser aplicado à qualidade do produto com o objetivo de identificar e fornecer informações sobre o nível de qualidade dos produtos durante o ciclo de vida do software. As métricas de progresso da qualidade propostas têm como objetivo verificar o nível de qualidade dos casos de uso do projeto e do projeto. As métricas definidas

avaliam o nível de qualidade de todos os casos de uso incorporados ao sistema, levando em consideração os casos de uso que estão em desenvolvimento e os já finalizados.

As métricas da visão de qualidade possuem as seguintes características: dirigida a casos de uso; baseada na inspeção de artefatos; e facilidade de observação (Meneses 2001). Neste sentido, a qualidade do software deriva diretamente da qualidade de cada caso de uso e a qualidade do caso de uso dá-se-a pela avaliação dos critérios de qualidade estabelecidos no plano de qualidade da organização. A métrica definida apresenta um valor único, que indica o percentual de qualidade obtida, facilitando e objetivando o entendimento, comum, a todos os *stakeholders* envolvidos.

O resultado (valor) de cada métrica, deve ser apresentado na escala de 0 (zero) a 1 (um), onde 0 (zero) indica que o caso de uso ou projeto não obteve nenhum nível de qualidade e 1 (um) indica que o caso de uso ou projeto obteve o valor máximo. Outra forma possível de representação é de forma percentual, sendo assim basta transformar, multiplicar, o valor obtido por cem.

3.5.1. Qualidade dos Artefatos

A qualidade do artefato é definida pelo conjunto dos itens do *checklist* de avaliação da qualidade. A qualidade do artefato, $\varphi_x(\text{casodeuso})$, é resultante da média aritmética dos itens atendidos satisfatoriamente pelo total de itens (atendidos e não atendidos) do *checklist* elaborado para avaliar o nível de qualidade do artefato. Neste definiu-se a temos a seguinte equação:

$$\varphi_x(\text{casodeuso}) = \frac{\sum_{x \in A} \text{Itens}_{\text{atendidos}}}{\sum_{x \in A} \text{Itens}_{\text{atendidos} + \text{não atendidos}}}$$

onde x pertence ao conjunto de artefatos A e $\varphi_x(\text{casodeuso})$ indica o nível de qualidade do artefato x para construção do *casodeuso* do projeto. Cada item deve ser descrito de forma clara e objetiva, evitando o surgimento de dúvidas e ambigüidade dos itens para cada artefato.

3.5.2. Qualidade do Caso de Uso

A métrica relacionada ao progresso da qualidade do caso de uso considera as fases do ciclo de vida, para obter uma visão mais detalhada do nível de qualidade do caso de uso na iteração ou fase, tornando mais fácil o acompanhamento da qualidade e permitindo uma ação corretiva, se necessário. Baseado em (Meneses 2001) temos a seguinte equação:

$$\varphi_{casodeuso}(i, f) = \frac{\sum_{x \in A_{if}} (p_x(casodeuso) \times \varphi_x(casodeuso))}{\sum_{x \in A_{if}} p_x(casodeuso)}$$

onde $p_x(casodeuso)$ indica o peso do artefato x para construção de $casodeuso$, $\varphi_x(casodeuso)$ representa a métrica de qualidade e indica percentualmente o nível de qualidade do artefato x produzido ou alterado na fase f do desenvolvimento do caso de uso, e A_{if} representa o conjunto de artefatos que devem ser produzidos ou atualizados na iteração i e fase f . O conjunto de artefatos a serem produzidos na iteração ou fase é definido pela organização. O valor de $\varphi_x(casodeuso)$ é obtido após as inspeções dos artefatos produzidos, resultante da media aritmética dos itens avaliados positivamente em relação ao total de itens para o artefato.

3.5.3. Qualidade do Projeto

Baseado em (Meneses 2001) temos a seguinte equação, para avaliar o progresso da qualidade do projeto:

$$\varphi_{projeto}(i) = \frac{\sum_{f \in FASES} (p_{casodeuso}(i, f) \times \varphi_{casodeuso}(i, f))}{\sum_{f \in FASES} p_{casodeuso}(i, f)}$$

onde i indica a iteração avaliada, f representa a fase do desenvolvimento do caso de uso avaliado, $p_{casodeuso}(i, f)$ indica o peso relativo ao caso de uso na iteração i e fase de

desenvolvimento f , $\varphi_{casodeuso}(i, f)$ indica o progresso da qualidade do caso de uso na iteração i e fase de desenvolvimento f , e $FASES$ corresponde às fases necessárias para o desenvolvimento do caso de uso. As fases adotadas pelo RUP são: $FASES = \{concepção, elaboração, construção, transição\}$.

Um peso foi associado ao caso de uso, pois os casos de uso possuem níveis de complexidade diferentes entre si e varia, também, durante as $FASES$ do projeto. Ou seja, geralmente é mais complexo codificar um caso de uso na $FASE$ de construção do que descrevê-lo na $FASE$ de concepção, a exemplo.

3.6. Exemplo da Obtenção da Visão de Qualidade

Para obter a visão de qualidade, primeiramente, precisa obter o cálculo do progresso da qualidade do projeto em um determinado momento. O momento pode ser escolhido de acordo com a definição da gerência, após término de cada iteração, fase ou marco do projeto. A seguir, um exemplo de como encontrar o valor $\varphi_{sistema}$ para o projeto em questão. O exemplo será apresentado seguindo o fluxo de atividades definido no processo de avaliação de progresso da qualidade de software.

A primeira atividade é o planejamento da avaliação, onde são identificados os casos de uso que serão avaliados. Neste exemplo, os casos de uso a serem avaliados são:

- UC01 – Manter produto: o caso de uso possui como ator principal a gerência de compras da empresa e é responsável em incluir, alterar e excluir os dados básicos do produto;
- UC02 – Manter cliente: o caso de uso possui como ator principal o próprio cliente e a gerência de vendas da empresa que é responsável em incluir, alterar e excluir os dados básicos do cliente e do endereço do cliente;
- UC03 – Manter pedido: o caso de uso tem como ator principal o vendedor que é responsável em cadastrar todos os produtos solicitados pelo cliente, verificando o estoque de cada item do pedido, bem como data de entrega e forma de pagamento do pedido;

- UC04 – Consultar pedido: responsável em listar todos os pedidos de acordo com o filtro definido. Tem como ator principal o vendedor e a gerência de vendas; e,
- UC05 – Validar usuário: este caso de uso é responsável por liberar ou não o acesso do usuário as funcionalidades do sistema.

Em seguida, os casos de uso são analisados e avaliados para ganhar um peso, pois, os casos de uso apresentam níveis de complexidade diferentes para cada iteração ou fase do projeto. O Gerente de projetos deverá atribuir os pesos para cada caso de uso do projeto que fazem parte do escopo de avaliação.

Neste exemplo, o caso de uso manter produto (UC01) deverá receber um peso menor do que o caso de uso manter cliente (UC02). Ou seja, o UC02 possui maior complexidade do que UC01, devido aos dados de endereço estar implícitos aos dados do cliente, sendo necessário validar um maior número de dados. Neste contexto, definiram-se os seguintes pesos (complexidade) para os casos de uso:

- $P_{\text{manter produto}} = 1$
- $P_{\text{manter cliente}} = 2$
- $P_{\text{manter pedido}} = 4$
- $P_{\text{consultar pedido}} = 1$
- $P_{\text{validar usuário}} = 1$

O próximo passo é identificar os artefatos que serão produzidos durante as fases do projeto. Neste exemplo, utilizaremos as fases definidas pelo Processo Unificado da Rational $FASES = \{\text{concepção, elaboração, construção}\}$ e definiu-se que todas as fases possuem o mesmo peso no desenvolvimento dos casos de uso, ou seja, $f \in FASES \rightarrow p_f = 1$. Os artefatos a serem produzidos são exemplificados a seguir por fases:

$A_{concepção} = \{ termo\ de\ viabilidade, classificação\ do\ projeto, plano\ de\ projeto \}$

$A_{elaboração} = \{ diagrama\ de\ UML, documento\ de\ requisitos, especificação\ de\ UC \}$

$A_{construção} = \{ código\ fonte, testes\ unitários \}$

Identificados os artefatos a serem produzidos ou atualizados em cada fase do projeto de desenvolvimento de software, o próximo passo é planejar a avaliação. Cada artefato, durante sua elaboração, pode possuir um nível de complexidade (peso) diferente de outro artefato. Objetivando simplificar este exemplo, adotou-se o mesmo peso 1 (um) para todos os artefatos, ou seja, $\forall x \in A_f, f \in FASES \rightarrow p_x = 1$ para todos os artefatos independentes da fase onde o artefato seja produzido. O próximo passo é realizar as inspeções seguindo os critérios de avaliação da qualidade, apoiados pelo checklist de avaliação dos artefatos. Após a inspeção, os valores obtidos são atribuídos a $\varphi_{caso\ de\ uso}(x)$, onde $x \in A_f, f \in FASES$.

Neste exemplo, foram exemplificadas três avaliações de progresso da qualidade do projeto. A primeira avaliação foi realizada ao final do primeiro marco do projeto, ou seja, logo após a fase de concepção. A segunda avaliação foi realizada após o marco da fase de elaboração e a terceira avaliação ao termino da fase de construção. Enfatiza-se que os valores apresentados a seguir, são apenas para contextualizar o exemplo. A seguir são apresentados os resultados das avaliações realizadas por fases para os casos de uso.

3.6.1.1. Primeira Avaliação (Fase Concepção)

A Tabela 3, apresenta em detalhes os resultados da primeira avaliação. O escopo desta avaliação foram todos os cinco casos de uso do sistema. Observa-se que o artefato termo de viabilidade para o caso de uso UC01 – Manter produto obteve um nível de qualidade de 90%.

Supondo que o *checklist* de avaliação deste artefato termo de viabilidade possua dez itens, o valor é resultante da inspeção realizada neste artefato, indica que nove dos dez itens foram avaliados satisfatoriamente, ou seja, nove dos dez itens atendem aos requisitos especificados, ou seja, 90%. Para os casos de uso UC01 e UC03, observa-se que ambos obtiveram o mesmo nível de qualidade de 80%, reflexo da influencia do nível de qualidade do artefato Plano de Projeto.

Tabela 3 – Avaliação: Marco Concepção

Casos de Uso	$P_{casodeuso}$	Artefatos	$P_{artefato}$	$\varphi_{artefato}$
UC01 - Manter produto	1	Termo de viabilidade	1	0,90
		Classificação do projeto	1	1,00
		Plano de projeto	1	0,50
		$\varphi_{caso\ de\ uso}(UC01)$		
UC02 - Manter cliente	2	Termo de viabilidade	1	0,90
		Classificação do projeto	1	1,00
		Plano de projeto	1	0,95
		$\varphi_{caso\ de\ uso}(UC02)$		
UC03 - Manter pedido	4	Termo de viabilidade	1	0,90
		Classificação do projeto	1	0,90
		Plano de projeto	1	0,60
		$\varphi_{caso\ de\ uso}(UC03)$		
UC04 - Consultar pedido	1	Termo de viabilidade	1	1,00
		Classificação do projeto	1	1,00
		Plano de projeto	1	1,00
		$\varphi_{caso\ de\ uso}(UC04)$		
UC05 - Validar usuário	1	Termo de viabilidade	1	1,00
		Classificação do projeto	1	1,00
		Plano de projeto	1	1,00
		$\varphi_{caso\ de\ uso}(UC05)$		

A qualidade do sistema para fase de concepção pode ser calculada pela expressão abaixo, cujo resultado do nível de qualidade do sistema foi de 87%. O nível de qualidade do sistema foi influenciado pelo nível de qualidade obtido nos casos de uso, levando em consideração o nível de complexidade dos casos de uso para fase de concepção.

$$\varphi_{sistema}(concepção) = \frac{1 \times 0,80 + 2 \times 0,95 + 4 \times 0,80 + 1 \times 1,00 + 1 \times 1,00}{1 + 2 + 4 + 1 + 1} = 0,87$$

3.6.1.2. Segunda Avaliação (Fase Elaboração)

Nesta segunda avaliação, realizada no marco da fase de elaboração do projeto, observa-se que o nível de qualidade de todos os casos de uso diminuiu exceto o caso de uso UC05 que manteve o mesmo nível de qualidade da primeira avaliação. A Tabela 4 apresenta os níveis de qualidade obtidos para cada artefato e para cada caso de uso.

Tabela 4 – Avaliação: Marco Elaboração

Casos de Uso	$P_{casodeuso}$	Artefatos	$P_{artefato}$	$\varphi_{artefato}$
UC01 - Manter produto	1	Diagrama UML	1	0,65
		Doc. de Requisitos	1	0,80
		Especificação de UC	1	0,80
		$\varphi_{caso\ de\ uso}(UC01)$		0,75
UC02 - Manter cliente	2	Diagrama UML	1	0,90
		Doc. de Requisitos	1	1,00
		Especificação de UC	1	0,50
		$\varphi_{caso\ de\ uso}(UC02)$		0,80
UC03 - Manter pedido	4	Diagrama UML	1	0,80
		Doc. de Requisitos	1	0,60
		Especificação de UC	1	0,40
		$\varphi_{caso\ de\ uso}(UC03)$		0,60
UC04 - Consultar pedido	1	Diagrama UML	1	0,85
		Doc. de Requisitos	1	1,00
		Especificação de UC	1	1,00
		$\varphi_{caso\ de\ uso}(UC04)$		0,95
UC05 - Validar usuário	1	Diagrama UML	1	1,00
		Doc. de Requisitos	1	1,00
		Especificação de UC	1	1,00
		$\varphi_{caso\ de\ uso}(UC05)$		1,00

A qualidade do sistema para fase de elaboração deve ser calculada novamente de modo a aferir o novo nível de qualidade do sistema e avaliar o progresso obtido pela nova avaliação. Sendo assim, temos:

$$\varphi_{sistema}(elaboração) = \frac{1 \times 0,75 + 2 \times 0,80 + 4 \times 0,60 + 1 \times 0,95 + 1 \times 1,00}{1 + 2 + 4 + 1 + 1} = 0,75$$

A qualidade do sistema obtida nesta segunda avaliação foi de 75%. Comparando com o nível de qualidade da primeira avaliação houve um decréscimo de 12%. O nível de qualidade do sistema foi influenciado pelos níveis de qualidade dos casos de uso avaliados.

O gerente de projetos deve avaliar o impacto do resultado obtido nesta avaliação no projeto e na conformidade aos requisitos especificados. Investigar as possíveis falhas no cumprimento do processo ou na elaboração dos artefatos produzidos pela equipe técnica, objetivando manter o nível de qualidade do projeto.

3.6.1.3. Terceira Avaliação (Fase Construção)

Nesta última avaliação, realizada no marco da fase de construção do projeto, foram inspecionados os artefatos de código fonte e de testes unitários. O artefato código fontes pode ser inspecionado quanto ao padrão de codificação da linguagem de programação e o artefato de testes unitários pode ser inspecionado quanto ao sucesso dos testes que foram elaborados, salvos se forem previamente definidos e contemplados pelo *checklist* de avaliação de cada artefato em questão.

Manteve-se o mesmo escopo das demais avaliações, os cinco casos de uso, onde se pode observar na Tabela 5 os níveis de qualidade obtidos para cada artefato e para cada caso de uso. Propositamente, destaca-se o nível de qualidade do caso de uso UC03, onde não foi produzido nenhum artefato para este caso de uso. Neste caso, o nível de qualidade para os artefatos não produzidos é zero, pois não se pode inspecionar um artefato que não existe.

Tabela 5 – Avaliação: Marco Construção

Casos de Uso	$P_{casodeuso}$	Artefatos	$P_{artefato}$	$\varphi_{artefato}$
UC01 - Manter produto	1	Código fonte	1	0,90
		Testes unitários	1	0,30
		$\varphi_{caso de uso}(UC01)$		0,60
UC02 - Manter cliente	2	Código fonte	1	0,90
		Testes unitários	1	0,60

			$\varphi_{\text{caso de uso}}(UC02)$	0,75
UC03 - Manter pedido	4	Código fonte	1	0,00
		Testes unitários	1	0,00
		$\varphi_{\text{caso de uso}}(UC03)$		0,00
UC04 - Consultar pedido	1	Código fonte	1	1,00
		Testes unitários	1	1,00
		$\varphi_{\text{caso de uso}}(UC04)$		1,00
UC05 - Validar usuário	1	Código fonte	1	1,00
		Testes unitários	1	1,00
		$\varphi_{\text{caso de uso}}(UC05)$		1,00

A qualidade do sistema para fase de construção deve ser novamente calculada. Sendo assim, temos:

$$\varphi_{\text{sistema}}(\text{construção}) = \frac{1 \times 0,60 + 2 \times 0,75 + 4 \times 0,00 + 1 \times 1,00 + 1 \times 1,00}{1 + 2 + 4 + 1 + 1} = 0,45$$

3.6.1.4. Avaliação Geral

As avaliações requerem o exame de artefatos, pois os artefatos são considerados indícios da aplicação prática de um processo e compõem o produto de software. As três avaliações, apresentadas acima, focaram determinados intervalos de tempo. O primeiro intervalo no início do projeto até o marco da concepção. O segundo do marco da concepção até o marco da elaboração. Por fim, do marco da elaboração até o marco da construção. Em todos os intervalos de tempo, foram avaliados os níveis de qualidade de cada artefato, casos de uso e do sistema por fase individualmente.

A seguir será apresentado o nível de qualidade dos casos de uso durante todas as fases do projeto e o nível de qualidade final do sistema:

UC01 – Manter Produto

- $\varphi_{concepção}(Manter\ Produto) = 0,80$
- $\varphi_{elaboração}(Manter\ Produto) = 0,75$
- $\varphi_{construção}(Manter\ Produto) = 0,60$

Assim, a qualidade do caso de uso UC01 – Manter Produto pode ser facilmente calculado como segue:

$$\varphi_{Manter\ Produto} = \frac{1 \times 0,80 + 1 \times 0,75 + 1 \times 0,60}{3} = 0,71$$

Observe que está sendo analisando a qualidade do caso de uso manter produto ao nível de fase do projeto, não levando em consideração as iterações de cada fase.

UC02 – Manter Cliente

- $\varphi_{concepção}(Manter\ Cliente) = 0,95$
- $\varphi_{elaboração}(Manter\ Cliente) = 0,80$
- $\varphi_{construção}(Manter\ Cliente) = 0,75$

Assim, a qualidade do caso de uso UC02 – Manter Cliente pode ser facilmente calculado como segue:

$$\varphi_{Manter\ Cliente} = \frac{1 \times 0,95 + 1 \times 0,80 + 1 \times 0,75}{3} = 0,83$$

UC03 – Manter Pedido

- $\varphi_{concepção}(Manter\ Pedido) = 0,80$
- $\varphi_{elaboração}(Manter\ Pedido) = 0,60$
- $\varphi_{construção}(Manter\ Pedido) = 0,00$

Assim, a qualidade do caso de uso UC03 – Manter Pedido pode ser facilmente calculado como segue:

$$\varphi_{Manter\ Pedido} = \frac{1 \times 0,80 + 1 \times 0,60 + 1 \times 0,00}{3} = 0,46$$

Observe que na fase de construção do caso de uso UC03 – Manter Pedido o valor obtido foi zero, devido às atividades desta fase não terem sido inicializadas e a inspeção não executada resultando no decréscimo, significativo, da qualidade do caso de uso. Ou seja, o caso de uso teve um nível de qualidade menor em relação aos demais casos de uso.

UC04 – Consultar Pedido

- $\varphi_{concepção}(Consultar\ Pedido) = 1,00$
- $\varphi_{elaboração}(Consultar\ Pedido) = 0,95$
- $\varphi_{construção}(Consultar\ Pedido) = 1,00$

Assim, a qualidade do caso de uso UC04 – Consultar Pedido pode ser facilmente calculado como segue:

$$\varphi_{Consultar\ Pedido} = \frac{1 \times 1,00 + 1 \times 0,95 + 1 \times 1,00}{3} = 0,98$$

UC05 – Validar Usuário

- $\varphi_{concepção}(Validar\ Usuário) = 1,00$
- $\varphi_{elaboração}(Validar\ Usuário) = 1,00$
- $\varphi_{construção}(Validar\ Usuário) = 1,00$

Assim, a qualidade do caso de uso UC05 – Validar Usuário pode ser facilmente calculado como segue:

$$\varphi_{ValidarUsuário} = \frac{1 \times 1,00 + 1 \times 1,00 + 1 \times 1,00}{3} = 1$$

Calculado a qualidade de cada caso de uso, o progresso da qualidade do projeto deriva diretamente da qualidade de cada caso de uso, levando-se em consideração o nível de complexidade de cada caso de uso já atribuídos anteriormente. Desse modo tem-se:

$$\varphi_{sistema} = \frac{P_{UC01} \times \varphi_{UC01} + P_{UC02} \times \varphi_{UC02} + P_{UC03} \times \varphi_{UC03} + P_{UC04} \times \varphi_{UC04} + P_{UC05} \times \varphi_{UC05}}{P_{UC01} + P_{UC02} + P_{UC03} + P_{UC04} + P_{UC05}}$$

Onde, $\varphi_{sistema}$ representa a qualidade do sistema, P_{UC} representa o nível de complexidade de cada caso de uso no sistema e φ_{UC} representa o nível qualidade do caso de uso no sistema. Substituindo os valores, obtém:

$$\varphi_{sistema} = \frac{1 \times 0,71 + 2 \times 0,83 + 4 \times 0,46 + 1 \times 0,98 + 1 \times 1,00}{1 + 2 + 4 + 1 + 1} = 0,68$$

Esse resultado indica que o projeto obteve um nível de qualidade de 0,68 em uma escala de zero (0) a um (1), que representa, respectivamente, valor mínimo e o valor máximo. Transformando em porcentagem, obteve-se um progresso de 68%. De posse deste resultado, o gerente de projeto pode compará-lo com valores obtidos em avaliações anteriores e observar

tendências sobre o projeto. Outro ponto a ser observado, é que a qualidade do caso de uso UC03 – Manter Pedido está abaixo da média dos demais casos de uso do projeto, precisando de acompanhamento e monitoramento mais próximo do gerente de projetos.

3.7. Ferramenta Inspector Panel

O Inspector Panel é uma ferramenta de controle para acompanhamento e avaliação do progresso da qualidade de projetos de software que dá apoio a automação do processo estabelecido em (Meneses 2001). A primeira versão da ferramenta foi desenvolvida em 2008 por (Vicente 2008) e atualizada em 2010 visando atender e cobrir a visão de progresso da qualidade em projetos de software.

A primeira versão da ferramenta atende a visão de desempenho e a visão de funcionalidade do processo Inspector (Meneses 2001). Esta segunda versão abrange as duas visões anteriores, integrando-as com a visão de qualidade apresentada neste trabalho de dissertação. Entre os objetivos da construção da ferramenta destacam-se:

- Automatização do processo e redução de falha na coleta. Ou seja, com a ferramenta é possível a realização/automatização das atividades do processo;
- Rapidez e eficiência na coleta e no processamento de dados. Depois que os dados foram levantados e devidamente registrados na ferramenta, já se pode realizar uma avaliação a fim de verificar o nível de qualidade do projeto;
- Oferece suporte à tomada de decisão. Ou seja, a ferramenta pode ser utilizada a fim de conceber fatos relevantes para a gerência de projeto. A partir daí, a gerência têm dados suficientes e saberá tomar uma decisão mais precisa em relação ao projeto de software; e,
- Geração de tabelas de progresso e de indicadores visuais (gráficos).

O Inspector Panel dispõe de uma interface amigável e auto-explicativa com menus navegáveis, visando facilitar o seu uso por profissionais e empresas desenvolvedoras de software que precisem acompanhar a qualidade de seus produtos. A Figura 10, apresenta a tela de inspeção onde é definido o escopo, casos de uso, de cada avaliação.

inspector panel

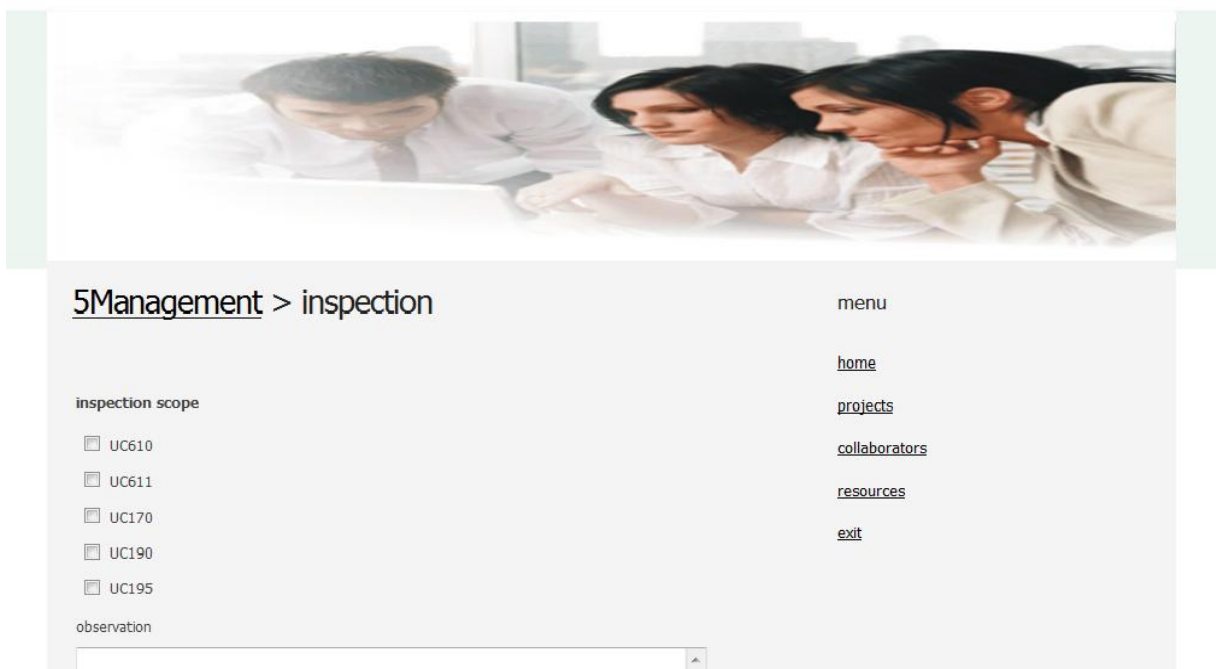


Figura 10. Inspector Panel: Tela de Inspeção

Para o ambiente do Inspector Panel foi definido e utilizado o NetBeans, adotando a linguagem Java, seguindo cuidadosamente as técnicas de orientação a objetos, objetivando de criar um produto com maior extensibilidade, reusabilidade e flexibilidade. Para a arquitetura, foram considerados fatores como finalidade do sistema, tipo de usuários e ambiente de execução. A arquitetura adotada no sistema é disposta em camadas independentes, e são apresentadas na Figura 11.

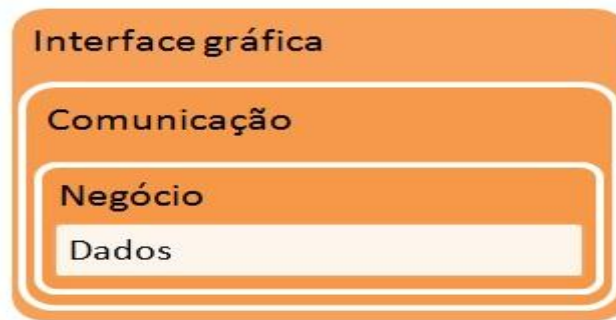


Figura 11. Arquitetura do Inspector Panel

Onde, a camada de Interface Gráfica é composta pelos componentes de apresentação da aplicação, responsáveis pela interação do usuário com o sistema; a camada de Comunicação implementa os mecanismos de comunicação entre a *Graphical User Interface* (GUI) e a camada de negócio. Isto torna o sistema independente da interface da qual ele faz uso no momento, facilitando a extensibilidade; a camada de Negócio implementa os serviços e as regras de negócio da aplicação; e por fim, a camada de Dados é responsável pelo armazenamento e recuperação dos dados do sistema.

Detalhes sobre as funcionalidades e uso da ferramenta Inspector Panel pode ser obtida em (Vicente 2008).

3.8. Considerações Finais

A aplicação do processo fornece informações sobre o andamento do nível de qualidade do projeto em desenvolvimento. **Monitorar o progresso da qualidade do projeto, podendo detalhar os resultados obtidos das inspeções de qualidade ao nível de fases, iterações e de casos de uso durante o ciclo de vida do software é o objetivo principal do processo.**

O presente capítulo apresentou alguns conceitos abordados pelo RUP. Definiu-se a visão de qualidade, seus os papéis e artefatos produzidos, fluxo de atividades foram documentados, seguidos de um exemplo para obter o progresso de qualidade. Estabeleceu as relações entre a qualidade dos artefatos, dos casos de uso e do projeto. Em seguida as métricas de qualidade de progresso dos casos de uso e do projeto foram definidas e baseadas em (Meneses 2001). Por fim, apresenta-se a ferramenta Inspector Panel desenvolvida para apoiar o processo de avaliação de progresso da qualidade, Inspector Q.

O próximo capítulo apresenta o estudo de caso realizado visando avaliar o nível de qualidade dos produtos de softwares que estão sendo desenvolvidos em uma secretaria do governo do estado de Pernambuco. Durante as avaliações foi utilizada como apoio, a ferramenta Inspector Panel. O Inspector Panel é responsável pela semi-automatização do processo de avaliação de progresso da qualidade apresentado neste trabalho.

4. ESTUDO DE CASO

*O lucro do nosso estudo é tornarmo-nos melhores e mais sábios.
—MICHEL DE MONTAIGNE*

Este capítulo objetiva apresentar um relato detalhado do estudo de caso realizado em uma organização pública do estado de Pernambuco, visando avaliar o progresso da qualidade em projetos de software dentro de uma organização.

Nas próximas seções são apresentadas as características particulares da organização, da equipe e do projeto avaliado. Em seguida, as avaliações de progresso qualidade, atividades realizadas e análise dos resultados obtidos. Posteriormente, as considerações finais são descritas visando à melhoria da qualidade em projetos de software na organização.

4.1. Introdução

A organização, onde foi realizado o estudo de caso, tem a missão de prover e gerir os recursos financeiros necessários à implementação de políticas públicas, apoiando o desenvolvimento econômico, social e cultural do estado de Pernambuco. A organização possui alguns valores, a exemplo de:

- Compromisso social – no sentido de atender às necessidades da população, saúde e educação pública;
- Equidade – tratamento equânime, igualitário e imparcial a todos, seja servidor público ou cidadão;
- Ética – comportamento segundo os princípios de austeridade, moralidade, probidade administrativa e respeito às pessoas em instituições, seja pública ou privada;
- Qualidade – prestação dos serviços de qualidade aos cidadãos; e,

- Transparência – nas ações e decisões tomadas, sobre as receitas e gestão dos gastos públicos com eficácia e eficiência visando à racionalização e melhoria da qualidade dos gastos.

Baseado na missão organizacional e nos valores adotados pela mesma, o Departamento de Tecnologia da Informação prioriza a busca pela agilidade dos processos organizacionais e pela qualidade da informação, adotando como estratégia o uso de arquiteturas, caracterizando em: arquitetura de projetos (curto, médio e longo prazo), arquitetura tecnológica e arquitetura de sistemas.

Para que as arquiteturas fossem implantadas na organização, foi preciso contratar, através do processo de licitação, empresas especializadas para desenvolvimento e instalação dos sistemas informáticos componentes da arquitetura para atender as seguintes áreas do estado de Pernambuco: planejamento, tributário, financeiro, administrativo e de recursos humanos. Sendo assim, neste trabalho de dissertação, este grande projeto foi denominado de 5Management, representando o gerenciamento das cinco áreas contempladas pelo projeto para atender o estado de Pernambuco.

4.2. Características do Ambiente e do Projeto

O ambiente do projeto é composto por uma estrutura física e de recursos tecnológicos, apoiados por ferramentas que auxiliam na captura de informações descritas pelo cliente. Neste ambiente é possível realizar reuniões de negócio com os clientes e reuniões do projeto com a equipe técnica. O ambiente é fortemente classificado como *outsourcing*, onde a organização obtém mão-de-obra de fora da empresa, ou seja, mão-de-obra terceirizada.

O *outsourcing* está ligado à idéia de subcontratação de serviços, onde é feita uma transferência das atividades para uma empresa terceirizada especializada. Essas atividades são conhecidas como atividades meio, e nunca como as atividades fins da organização.

A subcontratação é proveniente de uma seleção pública, licitação, entre as empresas desenvolvedoras de soluções de tecnologia da informação que estão devidamente registradas e aptas ao processo seletivo, seguindo as normas e critérios estabelecidos na licitação. Atualmente, existem diversas empresas juntas em comum acordo legal prestando serviços a

esta Secretaria. Este conjunto de empresas formam um consórcio visando o interesse comum de ganhar a licitação.

O projeto 5Management, concebido em junho de 2001, foi dividido em projetos menores visando um acompanhamento próximo dos objetivos a serem alcançados. Neste sentido, a área de planejamento, tributário, financeiro, administrativo e de recursos humanos ficou, respectivamente, com **quatro, nove, nove, dezesseis, cinco projetos de software**. O estudo de caso focou em um dos projetos da área tributária, onde foi possível realizar três avaliações para acompanhar o progresso da qualidade do projeto.

O projeto, avaliado, da área tributária tem como objetivo principal planejar, programar e distribuir as demandas internas (departamentos) e externas (sociedade) solicitadas, em cumprimento à legislação tributária vigente. Cada demanda é protocolada e pode resultar em uma ação de inclusão ou alteração ou verificação de informações. O trâmite de cada demanda aberta pode ser acompanhado por este sistema na web. O sistema é dividido em dezessete módulos, dos quais quatro ainda não foram desenvolvidos.

O estudo de caso focalizou em medir o progresso da qualidade de cinco casos de uso do projeto. A equipe do projeto é composta pelo líder da equipe que possui certificado *Project Management Professional* (PMP), por um analista de sistemas e seis desenvolvedores (contratados e estagiários). Para os desenvolvedores contratados são requeridos graduação em ciência ou engenharia da computação ou curso equivalente. A certificação Java é um grande diferencial para este profissional.

O projeto avaliado está baseado na arquitetura cliente-servidor, utiliza a linguagem de programação Java, apresenta as informações utilizando o *Java Server Pages* (JSP), tecnologia utilizada no desenvolvimento de aplicações para web. Como ferramenta de geração de relatórios utiliza-se o *Jasper Report* e disponibiliza alguns serviços, *webservices*, para uso interno da própria organização.

4.3. Metodologia do Estudo de Caso

O estudo de caso é classificado como um método de estudo de caso enquadrando-se como uma abordagem qualitativa. Segundo Oliveira (1999), o método é um conjunto de

processos pelo qual é possível conhecer uma determinada realidade, produzir um determinado objeto, desenvolver certos procedimentos ou comportamentos.

Yin (2001) afirma que os estudos de caso vão além das estratégias explanatórias, reforçando a possibilidade de estudos de caso como exploratórios ou descritivos. A realização do estudo de caso envolve três fases distintas:

- Definição do referencial teórico, seleção dos casos e o desenvolvimento de protocolos para coleta dos dados;
- Condução do estudo de caso, realizando a coleta de dados e analisando as informações obtidas; e, por fim
- Análise dos dados segundo o referencial teórico.

O estudo de caso teve início no final de agosto de 2010 e foi finalizado em dezembro do mesmo ano. Na primeira fase da condução do estudo de caso, foi iniciado com uma reunião junto à equipe de qualidade da organização, onde foi apresentado o objetivo do estudo de caso e os resultados a serem alcançados durante o mesmo. Sendo assim, o objetivo é avaliar o nível de qualidade dos projetos de software e como resultados espera-se apresentar os casos de uso que estão com o nível de qualidade abaixo da média, durante as fases ou iterações do projeto.

Na reunião, ficou definido que a realização do estudo de caso envolveria dois projetos, um da área tributária e o outro da área financeira. Porém, não foi possível realizar o acompanhamento do projeto da área financeira, devido às restrições de tempo. O *checklist* de inspeção dos artefatos do projeto, elaborado pela equipe de qualidade, foi o protocolo definido para coleta de dados. Após a coleta e inspeções, os dados foram registrados na ferramenta Inspector Panel.

Na segunda fase do estudo de caso foram colhidos os dados e armazenados em documentos formais elaborados pela equipe. Na terceira e última fase do estudo de caso, foram realizadas análises dos dados obtidos nas avaliações. A seguir, apresenta-se o detalhamento do procedimento das avaliações de progresso da qualidade.

4.4. Avaliação do Progresso da Qualidade

As avaliações são conduzidas pelo engenheiro de qualidade, desempenhando o papel de coletor de informações do processo, e tem como objetivo verificar a conformidade das atividades realizadas no projeto com os padrões e procedimentos estabelecidos na metodologia de desenvolvimento de software, bem como identificar oportunidades de melhoria na metodologia utilizada. Cada engenheiro de qualidade é responsável em inspecionar vários projetos na organização.

Neste estudo de caso, definiu-se que os valores da complexidade pode variar de 1 (um) a 10 (dez), onde o valor 1 (um) representa que o caso de uso ou artefato possui baixa complexidade e o valor 10 (dez) representa o maior nível de complexidade.

A avaliação é realizada através de entrevistas com o analista da equipe técnica do projeto e análise dos artefatos produzidos pela equipe técnica, com o apoio dos *checklists*. Foi realizada uma entrevista com o analista responsável pelo projeto e esclarecidos alguns pontos com o gerente do projeto. O papel do gerente de projetos foi abordado na Subseção 3.2.1.1.

Os dados e as informações são coletados pelo coletor de informações que é responsável em inspecionar e analisar os artefatos produzidos pela equipe. O papel do coletor de informações foi apresentado na Subseção 3.2.1.2.

Durante o projeto todos os artefatos são armazenados e disponibilizados em uma área comum da rede, facilitando o acompanhamento do projeto e toda a sua documentação. As inspeções dos artefatos eram realizadas, pelo coletor de informações, por etapas de acordo com a fase ou iteração avaliada. Cada etapa correspondia à inspeção de um conjunto de artefatos de uma determinada disciplina da metodologia de desenvolvimento da organização.

As subseções seguintes descrevem, em detalhes, as atividades realizadas durante o estudo de caso. Além disso, serão apresentadas as dificuldades que foram superadas, os resultados obtidos e algumas considerações importantes observados durante a aplicação do processo de avaliação de progresso da qualidade.

4.4.1. Atividades Realizadas

As atividades realizadas durante o estudo de caso são apresentadas conforme definição do fluxo de atividade do processo Inspector Q, apresentado neste trabalho. Deste modo, temos as seguintes atividades realizadas: planejamento das avaliações, coleta e processamento dos dados de qualidade, avaliação dos resultados e, por fim, encontrar soluções para os problemas encontrados.

4.4.1.1. Planejar a avaliação

O planejamento da avaliação de progresso da qualidade se dá pela definição da data da avaliação; pela identificação do escopo da avaliação, ou seja, quais os casos de uso que serão inspecionados; e pela identificação dos responsáveis pelas inspeções dos casos de uso.

Neste estudo de caso observou-se que a data de avaliação não obedece a nenhum padrão ou critério estabelecido. A data de avaliação é definida de acordo com as prioridades da equipe de qualidade, sendo assim, após o término de uma iteração ou fase do projeto (marco), é possível que a iteração ou fase não seja avaliada de imediato. Evidentemente é importante que as avaliações sejam programadas para o final de cada marco (iteração ou fase) do projeto, pois se compara os resultados obtidos com o planejado, sendo possível tomar decisões, a tempo, para corrigir atividades que prejudiquem a qualidade do produto final.

O processo de avaliação de progresso da qualidade é executado por um conjunto lógico e seqüencial de atividades descritos no Capítulo 3. Como em toda atividade, se faz necessário estabelecer prioridades para sua execução. Desta forma, as atividades do processo de avaliação são adicionadas ao conjunto de outras atividades da equipe de qualidade, sendo necessário restabelecer as prioridades das atividades da equipe de qualidade.

Quanto ao escopo de avaliação, todos os casos de uso do módulo que está sendo desenvolvido no projeto foram inspecionados, de modo a aferir o nível de qualidade do módulo que será incorporado ao produto de software. Para cada projeto é designado apenas um engenheiro de qualidade para realizar as inspeções dos artefatos e proceder com a avaliação. Posteriormente, o engenheiro de qualidade comunica a todos os interessados o resultado da avaliação.

4.4.1.2. Coletar e processar dados da qualidade

Nesta atividade são coletados e processados os dados da qualidade dos casos de uso que estão dentro do escopo de avaliação. A atividade de coleta e processamento visa medir o nível de qualidade dos casos de uso do projeto, através de inspeções realizadas nos artefatos produzidos.

As inspeções dos artefatos são realizadas pelo coletor de informações e deve obedecer aos critérios definidos no artefato critérios de avaliação da qualidade dos artefatos. Os critérios definidos foram: concisão, consistência, facilidade de compreensão, legibilidade, integridade, conformidade, não ambigüidade.

As inspeções são documentadas com o apoio dos artefatos elaborados pela equipe de qualidade, os *checklists*. Durante o estudo de caso foi possível realizar três avaliações, o escopo dessas avaliações foram todos os casos de uso do módulo do projeto avaliado. Não foi possível realizar mais avaliações devido a alguns atrasos no cumprimento dos prazos estabelecidos pelo projeto, resultantes da instabilidade dos requisitos e desfalque de membros da equipe.

Cada módulo é considerado como um mini-projeto a ser incorporado no projeto principal, seguindo a metodologia de desenvolvimento de software estabelecida na organização. Cada avaliação representa a avaliação do módulo em desenvolvimento, ou seja, as três avaliações representam a evolução do nível de qualidade do módulo desenvolvido.

As fases definidas pela metodologia da organização são as mesmas definidas pelo RUP, sendo assim, as fases são: *FASES = {concepção, elaboração, construção, transição}*. A seguir serão apresentadas, em detalhes, as avaliações realizadas pela equipe de qualidade.

4.4.1.2.1. Primeira Avaliação

A primeira avaliação utilizando o processo de avaliação de progresso de qualidade foi considerada um desafio, pois a inclusão de novas atividades em uma metodologia definida e madura dentro de em um contexto organizacional, pode gerar expectativas e receios de ambos os lados da avaliação, o lado do avaliador (coletor de informações) e do avaliado (equipe do

projeto). O nível de qualidade obtida na primeira avaliação, realizada no dia 20 de setembro de 2010, foi $\varphi_{projeto} = 0,86$.

No momento da primeira avaliação, o projeto já estava em andamento e encontrava-se no final da fase de construção e início da fase de transição. Sendo assim, foi necessário realizar um levantamento dos dados ao longo das fases anteriores e, posteriormente, simular a avaliação de progresso da qualidade no projeto durante as fases.

Fizeram parte do escopo da primeira avaliação os casos de uso UC610 e UC611. Ambos fazem parte do mesmo módulo e são complementares. O UC611 só pode ser executado posteriormente a execução do UC610, obedecendo à regra de negócio estabelecida pelos *stakeholders*. Os módulos do projeto são estabelecidos para cada objetivo do projeto, não levando em consideração a quantidade de casos de uso gerados, o tempo e o esforço necessário para construção do novo objetivo. Pode-se citar como exemplo de um objetivo, o módulo de gerenciamento de vendas e, este, possui cinco casos de uso.

O nível de complexidade (peso) dos casos de uso foi definido pelo analista de negócio do projeto. Nesta avaliação ambos os casos de uso possuem o mesmo nível de complexidade devido à similaridade e proximidade dos casos de uso. Os pesos definidos foram $p_{UC\ 610} = 8$ e $p_{UC\ 611} = 8$. Posteriormente, foram definidos os pesos dos casos de uso por fases do projeto, obtendo os seguintes valores para os casos de uso UC610 e UC611:

- $UC's = \{concepção = 6, elaboração = 6, construção = 8, transição = 8\}$

Nesta avaliação observou-se a existência de apenas dois casos de uso, podendo avaliar os resultados obtidos na avaliação através da Tabela 6 e da Tabela 7. A primeira tabela apresenta a qualidade dos casos de uso por iteração e na segunda tabela é apresentada a qualidade dos casos de uso por fases do projeto. Em ambas tabelas observa-se que para cada fase do projeto teve apenas uma iteração e que o nível de qualidade do UC611 supera o nível de qualidade do UC610 nas primeiras iterações.

Tabela 6 – Primeira Avaliação: Qualidade dos Casos de Uso por Iterações

Data da Avaliação	Iteração	Caso de Uso	$P_{casodeuso}$	$\varphi_{casodeuso}$
20/09/2010	1º Iteração	UC610	6	0,45
		UC611	6	1,00
		$\varphi_{(1^{\circ} \text{ Iteração})}$		0,73
	2º Iteração	UC610	6	0,61
		UC611	6	0,89
		$\varphi_{(2^{\circ} \text{ Iteração})}$		0,75
	3º Iteração	UC610	8	0,90
		UC611	8	0,95
		$\varphi_{(3^{\circ} \text{ Iteração})}$		0,93
	4º Iteração	UC610	8	1,00
		UC611	8	1,00
		$\varphi_{(4^{\circ} \text{ Iteração})}$		1,00
				$\varphi_{projeto}$

Tabela 7 – Primeira Avaliação: Qualidade dos Casos de Uso por Fase

Data da Avaliação	Fase	Caso de Uso	$P_{casodeuso}$	$\varphi_{casodeuso}$
20/09/2010	Concepção	UC610	6	0,45
		UC611	6	1,00
		$\varphi_{(concepção)}$		0,73
	Elaboração	UC610	6	0,61
		UC611	6	0,89
		$\varphi_{(elaboração)}$		0,75
	Construção	UC610	8	0,90
		UC611	8	0,95
		$\varphi_{(construção)}$		0,93
	Transição	UC610	8	1,00
		UC611	8	1,00
		$\varphi_{(transição)}$		1,00
				$\varphi_{projeto}$

Na fase de concepção o UC610 obteve um nível de qualidade inferior a 50%, isto devido à falta de informações nos artefatos gerados para o caso de uso, incluindo a não elaboração ou geração de alguns artefatos importantes definidos pela metodologia de desenvolvimento de software da organização.

Embora a falta de artefatos indiquem uma violação no processo de desenvolvimento do produto, gerando grandes riscos à qualidade do produto final, não é impeditivo para que o módulo seja entregue aos *stakeholders*. Para os *stakeholders* o módulo do software pronto e executando da forma esperada é mais importante do que a documentação. Da mesma forma, a fase de elaboração teve problemas semelhantes à fase anterior com uma melhoria na qualidade dos artefatos produzidos para o UC610 e uma queda do nível de qualidade para UC611.

A fase de construção é abordada em duas etapas. A primeira etapa compreendeu a implementação do código fonte dos casos de uso e a segunda no planejamento e execução dos testes unitários. Os casos de uso UC610 e UC611 obtiveram níveis de qualidade altos com uma diferença de 0,05 ou de 5,00% entre eles. Devido à falta de registros de informações dos testes executados que obtiveram sucesso ou insucesso e o não cumprimento de todas as atividades, nesta fase, os casos de uso não obtiveram valor máximo nesta avaliação.

Pode-se observar que a fase de transição ou a quinta iteração do módulo do projeto, obteve o maior nível de qualidade, ou seja, 100%. O bom nível de qualidade, nesta fase, é resultado de um processo de transição definido e executado dentro dos padrões estabelecidos na organização. A fase de transição é contextualizada da seguinte forma: (i) pelo procedimento de atualização da documentação, registrando as novas funcionalidades ou alterações de funcionalidades do software; e (ii) pelo procedimento de instalar ou atualizar os códigos fontes no ambiente de homologação do usuário. Posteriormente, após homologação segue para o ambiente de produção se o cliente assim desejar. Nesta fase a qualidade obtém o

valor máximo devido a um script *Shell*³ que é responsável em executar toda sequência definida e por realizar toda a validação nos códigos fontes.

Na validação, os fontes são inspecionados com o objetivo de identificar possíveis *bugs* de programação utilizando o *FindBug*, o PMD e o *Copy/ Paste Detector* (CPD). A organização estava certa que com a adoção do *FindBug* e PMD no projeto podiam reduzir os erros no ambiente de produção, adotou-os de forma impeditivas e não negociadas, ou seja, não é possível realizar a transição caso seja encontrado algum registro de *FindBug* ou PMD. Os registros de CPD são considerados opcionais e são avaliados pela equipe de desenvolvimento.

Para obter os níveis de qualidade que cada caso de uso atingiu no projeto, aplica-se a seguinte equação como segue:

$$\varphi_{UC\ 610} = \frac{6 \times 0,45 + 6 \times 0,61 + 8 \times 0,90 + 8 \times 1,00}{6 + 6 + 8 + 8} = 0,77$$

$$\varphi_{UC\ 611} = \frac{6 \times 1,00 + 6 \times 0,89 + 8 \times 0,95 + 8 \times 1,00}{6 + 6 + 8 + 8} = 0,96$$

O nível de qualidade do projeto é influenciado pelo níveis de qualidade dos casos de uso do projeto. Sendo assim, o nível de qualidade do projeto é definido pela seguinte equação:

$$\varphi_{projeto} = \frac{P_{UC\ 610} \times \varphi_{UC\ 610} + P_{UC\ 611} \times \varphi_{UC\ 611}}{P_{UC\ 610} + P_{UC\ 611}}$$

onde, $\varphi_{projeto}$ indica o nível de qualidade do projeto, P_{UC} representa a complexidade que o caso de uso tem para o projeto e φ_{UC} representa o nível de qualidade do caso de uso obtido pela avaliação no projeto. Substituindo os valores correspondentes temos:

³ Linguagem de script usada em vários sistemas operacionais, com diferentes dialetos, dependendo do interpretador de comandos utilizado.

$$\varphi_{projeto} = \frac{8 \times 0,77 + 8 \times 0,96}{8 + 8} = 0,86$$

4.4.1.2.2. Segunda Avaliação

A qualidade obtida na segunda avaliação, realizada no dia 25 de outubro de 2010, foi $\varphi_{sistema} = 0,80$. Fizeram parte do escopo da segunda avaliação os seguintes casos de uso UC610, UC611, UC170, UC190 e UC195. Os casos de uso UC610 e UC611, da primeira avaliação foram mantidos no escopo durante a segunda avaliação de modo a estabelecer um novo nível de qualidade para os casos de uso. Infelizmente, não houve um progresso da qualidade para os casos de uso UC610 e UC611, pois, mesmo com atividades corretivas estabelecidas pela equipe de qualidade, nenhuma destas atividades foram finalizadas para aumentar o nível de qualidade dos casos de uso. Sendo assim, os níveis de qualidade destes casos de uso continuam os mesmos não havendo um progresso do nível de qualidade.

O surgimento de novos casos de uso no projeto pode influenciar o nível de qualidade já adquirido pelo sistema. Esta influencia pode resultar em um acréscimo ou recuo do nível de qualidade do projeto, pois o nível de complexidade faz parte do denominador da equação apresentada na Subseção 3.5.3.

O analista de negócio do projeto estabeleceu os níveis de complexidade (peso) dos casos de uso. Nesta avaliação os casos de uso possuem níveis de complexidade diferentes variando num intervalo muito pequeno. Os pesos definidos foram $p_{UC\ 170} = 8$, $p_{UC\ 190} = 7$ e $p_{UC\ 195} = 8$. Os pesos dos casos de uso UC610 e UC611 não foram reavaliados, permanecendo igual à primeira avaliação, sendo: $p_{UC\ 610} = 8$ e $p_{UC\ 611} = 8$. Posteriormente, foram definidos os pesos dos casos de uso por fases do projeto, obtendo os seguintes valores para os casos de uso UC610, UC611, UC170, UC190 e UC195:

- $UC610 = \{concepção = 6, elaboração = 6, construção = 8, transição = 8\}$
- $UC611 = \{concepção = 6, elaboração = 6, construção = 8, transição = 8\}$

- $UC170 = \{concepção = 5, elaboração = 7, construção = 8, transição = 7\}$
- $UC190 = \{concepção = 7, elaboração = 7, construção = 7, transição = 7\}$
- $UC195 = \{concepção = 5, elaboração = 7, construção = 8, transição = 7\}$

Nesta avaliação foram avaliados cinco casos de uso, podendo avaliar os resultados obtidos na avaliação através da Tabela 8 e Tabela 9. A primeira tabela apresenta a qualidade dos casos de uso por iteração e na segunda tabela é apresentada a qualidade dos casos de uso por fases do projeto. Em ambas as tabelas, observa-se o nível de qualidade dos casos de uso e do projeto, sendo possível identificar quais os casos de uso que estão em dificuldades em atingir um bom nível de qualidade. Os casos de uso UC190 e UC195 obtiveram, aproximadamente, o mesmo nível de qualidade no projeto, sendo superior ao nível de qualidade obtido pelo UC170.

Tabela 8 – Segunda Avaliação: Qualidade dos Casos de Uso por Iteração

Data da Avaliação	Iteração	Caso de Uso	$P_{casodeuso}$	$\varphi_{casodeuso}$
25/10/2010	5º Iteração	UC610	6	0,45
		UC611	6	1,00
		UC170	5	1,00
		UC190	7	1,00
		UC195	5	1,00
		$\varphi_{(5^\circ \text{ Iteração})}$		0,89
	6º Iteração	UC610	6	0,61
		UC611	6	0,89
		UC170	7	0,48
		UC190	7	0,56
		UC195	7	0,66
		$\varphi_{(6^\circ \text{ Iteração})}$		0,63
	3º Iteração	UC610	8	0,90
		UC611	8	0,95
		UC170	8	-
		UC190	7	-
		UC195	8	-
		$\varphi_{(3^\circ \text{ Iteração})}$		0,93

	4º Iteração	UC610	8	1,00
		UC611	8	1,00
		UC170	7	-
		UC190	7	-
		UC195	7	-
				$\varphi_{projeto}$ 0,80

Na Tabela 8, observa-se o contexto da avaliação até sexta iteração presente até a fase de elaboração. Em seguida, foram mantidos os resultados da avaliação da terceira e quarta iteração, pois o instante (tempo) da avaliação ocorreu ao final da sexta iteração.

Tabela 9 – Segunda Avaliação: Qualidade dos Casos de Uso por Fases

Data da Avaliação	Fase	Caso de Uso	$P_{casodeuso}$	$\varphi_{casodeuso}$	
25/10/2010	Concepção	UC610	6	0,45	
		UC611	6	1,00	
		UC170	5	1,00	
		UC190	7	1,00	
		UC195	5	1,00	
					$\varphi_{(Concepção)}$
	Elaboração	UC610	6	0,61	
		UC611	6	0,89	
		UC170	7	0,48	
		UC190	7	0,56	
		UC195	7	0,66	
					$\varphi_{(Elaboração)}$
	Construção	UC610	8	0,90	
		UC611	8	0,95	
		UC170	8	-	
		UC190	7	-	
		UC195	8	-	
					$\varphi_{(Construção)}$
	Transição	UC610	8	1,00	
		UC611	8	1,00	
		UC170	7	-	

	UC190	7	-
	UC195	7	-
		$\varphi_{(Transição)}$	1,00
		$\varphi_{projeto}$	0,80

Analisando as duas avaliações realizadas, até o presente momento, observa-se uma redução do nível de qualidade do progresso total, onde na primeira avaliação o nível de qualidade obtido foi de $\varphi_{sistema} = 0,86$, contrapondo com os $\varphi_{sistema} = 0,80$ obtidos nesta avaliação. A diminuição do nível total da qualidade do projeto foi influenciada pelo nível de qualidade dos casos de uso incluídos no escopo da avaliação.

A princípio, observa-se claramente que a fase de elaboração merece uma atenção do gerente de projeto, visando identificar os motivos que levam o não atendimento de todos os critérios de qualidade estabelecidos. Certamente, caso seja necessário investir em treinamentos de análise, diagramas UML, entre outros, a gerencia do projeto deve estabelecer ações para que esta necessidade seja suprida, visando garantir a qualidade do projeto nesta fase e nas fases posteriores.

Para obter os níveis de qualidade de cada caso de uso em relação às fases do projeto, aplica-se a seguinte equação como segue:

$$\varphi_{UC\ 610} = \frac{6 \times 0,45 + 6 \times 0,61 + 8 \times 0,90 + 8 \times 1,00}{6 + 6 + 8 + 8} = 0,77$$

$$\varphi_{UC\ 611} = \frac{6 \times 1,00 + 6 \times 0,89 + 8 \times 0,95 + 8 \times 1,00}{6 + 6 + 8 + 8} = 0,96$$

$$\varphi_{UC\ 170} = \frac{5 \times 1,00 + 7 \times 0,48}{5 + 7} = 0,70$$

$$\varphi_{UC\ 190} = \frac{7 \times 1,00 + 7 \times 0,56}{7 + 7} = 0,78$$

$$\varphi_{UC\ 195} = \frac{5 \times 1,00 + 7 \times 0,66}{5 + 7} = 0,80$$

O nível de qualidade dos casos de uso $\varphi_{UC\ 170}$, $\varphi_{UC\ 190}$ e $\varphi_{UC\ 195}$ ficaram acima dos setenta por cento, resultante do nível de qualidade das fases de concepção e elaboração. Nesta avaliação as fases de construção e transição não tinham sido inicializadas para os novos casos de uso. Sendo assim, o nível de qualidade dos casos de uso deve ser calculado em relação às fases ou iterações concluídas do projeto. Ou seja, fases ou iterações não finalizadas não podem fazer parte do cálculo do nível de qualidade, pois não se pode avaliar uma fase ou iteração inexistente.

O nível de qualidade do projeto é calculado a partir dos níveis de qualidade dos casos de uso que fizeram parte do escopo de avaliação. Desta forma, temos:

$$\varphi_{projeto} = \frac{8 \times 0,77 + 8 \times 0,96 + 8 \times 0,70 + 7 \times 0,78 + 8 \times 0,80}{8 + 8 + 8 + 7 + 8} = 0,80$$

Ao final de cada avaliação são apresentadas as pendências identificadas a todos os envolvidos. As pendências são registradas em uma ferramenta que gerencia todas as não conformidades do projeto e só podem ser finalizadas quando passarem por uma avaliação e aprovação da equipe de qualidade.

Para as próximas avaliações, o coletor de informações, sugeriu que toda alteração ou modificação no sistema seja tratada como um projeto que contemple todos os passos da metodologia de desenvolvimento da organização, o que facilitaria a elaboração do projeto como também a documentação.

4.4.1.2.3. Terceira Avaliação

A qualidade obtida na terceira e última avaliação foi $\varphi_{sistema} = 0,85$, ou seja, 85,79% mais precisamente. Fizeram parte do escopo desta avaliação os mesmos casos de uso da segunda avaliação que são os seguintes: UC610, UC611, UC170, UC190 e UC195. Os casos de uso UC610 e UC611, presente deste a primeira avaliação foram mantidos no escopo de modo a estabelecer um novo nível de qualidade para os casos de uso e para o projeto.

Novamente e infelizmente, não houve um progresso da qualidade para os casos de uso UC610 e UC611. Nesta terceira avaliação, não houve inclusão de novos casos de uso, mas, foi estabelecida pequena alterações na regra de negócio a ser incorporada no módulo, contemplado pelos casos de uso UC170, UC190 e UC195.

Baseado nesta mudança de regra, o analista planejou novas iterações para as fases de concepção, elaboração e construção do projeto. A mudança na regra gerou no sistema um subfluxo para tratar exclusivamente do atendimento, prioritário, de cidadãos e de outros órgãos públicos que estejam em uma área de riscos ou de calamidade, resultante de um desastre natural.

O analista de negócio responsável pelo projeto mencionou que atualizar toda a documentação, fazendo o mapeamento adequado visando garantir a integridade das informações, associado à importância e representatividade do projeto na organização, tende a ser uma tarefa árdua. As fases de concepção, elaboração e transição abordam artefatos de documentação do projeto e por isso os níveis de complexidade dos casos de uso aproximam-se do nível de complexidade na fase de construção. Consideram-se artefatos de documentação do projeto todos os artefatos, exceto código fonte.

O analista de negócio do projeto estabeleceu (manteve) níveis de complexidade dos casos de uso, sendo os pesos: $p_{UC\ 610} = 8$, $p_{UC\ 611} = 8$, $p_{UC\ 170} = 8$, $p_{UC\ 190} = 7$ e $p_{UC\ 195} = 8$. Posteriormente, foram definidos os pesos dos casos de uso por fases do projeto, obtendo os seguintes valores para os casos de uso UC610, UC611, UC170, UC190 e UC195:

- $UC610 = \{concepção = 6, elaboração = 6, construção = 8, transição = 8\}$
- $UC611 = \{concepção = 6, elaboração = 6, construção = 8, transição = 8\}$
- $UC170 = \{concepção = 5, elaboração = 7, construção = 8, transição = 7\}$
- $UC190 = \{concepção = 7, elaboração = 7, construção = 7, transição = 7\}$
- $UC195 = \{concepção = 5, elaboração = 7, construção = 8, transição = 7\}$

O resultado desta avaliação pode ser visualizado através das seguintes tabelas: Tabela 10 e Tabela 11.

Tabela 10 – Terceira Avaliação: Qualidade dos Casos de Uso por Iterações

Data da Avaliação	Iteração	Caso de Uso	$P_{casodeuso}$	$\varphi_{casodeuso}$
11/11/2010	7º Iteração	UC610	6	0,45
		UC611	6	1,00
		UC170	5	1,00
		UC190	7	1,00
		UC195	5	1,00
		$\varphi_{(7^{\circ} \text{ Iteração})}$		0,89
	8º Iteração	UC610	6	0,61
		UC611	6	0,89
		UC170	7	0,88
		UC190	7	0,90
		UC195	7	0,91
		$\varphi_{(8^{\circ} \text{ Iteração})}$		0,84
	9º Iteração	UC610	8	0,90
		UC611	8	0,95
		UC170	8	0,83
		UC190	7	0,96
		UC195	8	0,90
		$\varphi_{(9^{\circ} \text{ Iteração})}$		0,91
	4º Iteração	UC610	8	1,00
		UC611	8	1,00
		UC170	7	-
		UC190	7	-
		UC195	7	-
		$\varphi_{(4^{\circ} \text{ Iteração})}$		1,00
			$\varphi_{projeto}$	0,90

Tabela 11 – Terceira Avaliação: Qualidade dos Casos de Uso por Fase

Data da Avaliação	Fase	Caso de Uso	$P_{casodeuso}$	$\varphi_{casodeuso}$
11/11/2010	Concepção	UC610	6	0,45
		UC611	6	1,00

	UC170	5	1,00
	UC190	7	1,00
	UC195	5	1,00
	$\varphi_{(Concepção)}$		0,89
Elaboração	UC610	6	0,61
	UC611	6	0,89
	UC170	7	0,88
	UC190	7	0,90
	UC195	7	0,91
	$\varphi_{(Elaboração)}$		0,84
Construção	UC610	8	0,90
	UC611	8	0,95
	UC170	8	0,83
	UC190	7	0,96
	UC195	8	0,90
	$\varphi_{(Construção)}$		0,91
Transição	UC610	8	1,00
	UC611	8	1,00
	UC170	7	-
	UC190	7	-
	UC195	7	-
	$\varphi_{(Transição)}$		1,00
$\varphi_{projeto}$			0,90

Nesta terceira avaliação, o caso de uso UC170 obteve um incremento significativo do nível de qualidade resultando em $\varphi_{UC170} = 0,88$ em contraponto com o nível da segunda avaliação que era de $\varphi_{UC170} = 0,48$. Da mesma forma, os casos de uso UC190 e UC195 obtiveram níveis de qualidade maiores em relação à segunda avaliação, onde os níveis de qualidade dos casos de uso na terceira avaliação foram de $\varphi_{UC190} = 0,90$ e $\varphi_{UC195} = 0,91$. Os níveis de qualidade da segunda avaliação (avaliação anterior) foram de $\varphi_{UC190} = 0,56$ e $\varphi_{UC195} = 0,66$.

Analisando pode-se observar que houve uma evolução do nível de qualidade dos casos de uso UC170, UC190 e UC195 em todas as fases do projeto, com exceção da fase de transição que se manteve inalterável. A evolução foi reflexo de um acompanhamento e correções dos desvios encontrados. Alguns exemplos de desvios que encontrados e corrigidos foram: (1) atualização do documento de requisitos; (2) elaboração do termo de homologação de requisitos; (3) inclusão do nome do aprovador das regras de negócios; e, (4) elaboração do plano de testes.

Para obter os níveis de qualidade de cada caso de uso, aplica-se a métricas definidas para determinar o nível de qualidade dos casos de uso, a equação é apresentada a seguir:

- $$\varphi_{UC\ 610} = \frac{6 \times 0,45 + 6 \times 0,61 + 8 \times 0,90 + 8 \times 1,00}{6 + 6 + 8 + 8} = 0,77$$

- $$\varphi_{UC\ 611} = \frac{6 \times 1,00 + 6 \times 0,89 + 8 \times 0,95 + 8 \times 1,00}{6 + 6 + 8 + 8} = 0,96$$

- $$\varphi_{UC\ 170} = \frac{5 \times 1,00 + 7 \times 0,88 + 8 \times 0,83}{5 + 7 + 8} = 0,89$$

- $$\varphi_{UC\ 190} = \frac{7 \times 1,00 + 7 \times 0,90 + 7 \times 0,96}{7 + 7 + 7} = 0,95$$

- $$\varphi_{UC\ 195} = \frac{5 \times 1,00 + 7 \times 0,91 + 8 \times 0,90}{5 + 7 + 8} = 0,93$$

O nível de qualidade do projeto é calculado a partir dos níveis de qualidade dos casos de uso que fizeram parte do escopo de avaliação. Desta forma, temos:

$$\varphi_{projeto} = \frac{8 \times 0,77 + 8 \times 0,96 + 8 \times 0,89 + 7 \times 0,95 + 8 \times 0,93}{8 + 8 + 8 + 7 + 8} = 0,90$$

O incremento do nível de qualidade do projeto foi de apenas 0,10. Este progresso de qualidade do projeto é reflexo do aumento do progresso de qualidade dos casos de uso UC170, UC190 e UC195.

4.4.1.3. Avaliar os resultados

As avaliações de progresso da qualidade serviram para identificar os casos de uso com baixo nível de qualidade e para determinar o progresso de qualidade do projeto obtido durante essas avaliações. Durante as avaliações foram levados em consideração os artefatos que deveriam ser produzidos e por quaisquer motivos (tempo, esforço, etc) não foram produzidos. Sendo assim, os artefatos não produzidos receberam o valor mais baixo, zero.

Apresentam-se os níveis de qualidade dos casos de uso, na Figura 12, obtidos durante a primeira, segunda e terceira avaliações. Nesta figura observa-se que inicialmente o nível de qualidade dos casos de uso são zero, pois nenhuma inspeção foi realizada.

Na primeira avaliação apenas os casos de uso UC610 e UC611 fizeram parte do escopo de avaliação, onde o UC610 obteve um nível de qualidade inferior em relação ao UC611. O fato é que alguns artefatos referentes ao UC611 não foram produzidos, baixando o nível de qualidade do caso de uso.

Manteve-se o mesmo nível de qualidade dos casos de uso UC610 e UC611 durante as demais avaliações, pois não foram desempenhadas atividades corretivas visando melhorar a qualidade dos casos de uso em questão. Enfatiza-se que cada organização deve determinar o nível de qualidade na aceitação para cada caso de uso. Durante o estudo de caso, todos os casos de uso foram aceitos, homologados, pelos *stakeholders* (clientes).

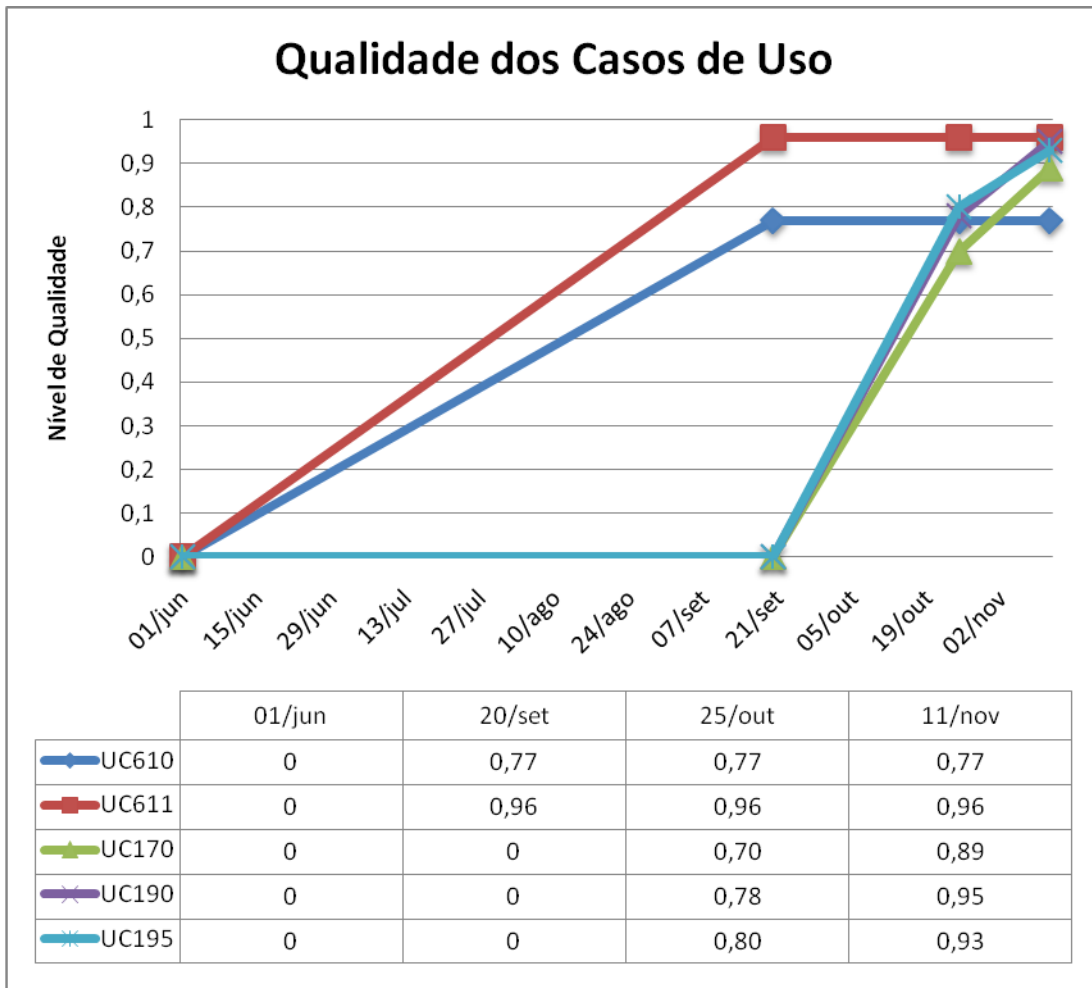


Figura 12. Qualidade dos UC's *versus* Avaliações

Na segunda avaliação os casos de uso UC190 e UC195 obtiveram, praticamente, os mesmos níveis de qualidade, pois apresentam dependência e familiaridade entre si no projeto. O nível de qualidade do UC170 foi de $\varphi_{UC\ 170} = 0,70$ resultante da identificação de falhas no cumprimento das atividades da metodologia de software adotada pela organização. Na terceira avaliação, constatou-se um progresso do nível de qualidade para todos os casos de uso, exceto para os casos de uso UC610 e UC611.

Destaca-se um significativo progresso de qualidade para o caso de uso UC170, onde uma nova iteração com atividades foi planejada, elaborada e desenvolvida, contemplando os casos de uso que tiveram progresso de qualidade, ou seja, os casos de uso UC170, UC190 e UC195.

A qualidade dos casos de uso contribui para melhorar a qualidade do projeto, ou seja, o nível de qualidade do projeto deriva diretamente dos níveis de qualidade dos casos de uso. A Figura 13, apresenta os níveis de qualidade do projeto obtidos durante a primeira, segunda e terceira avaliação.

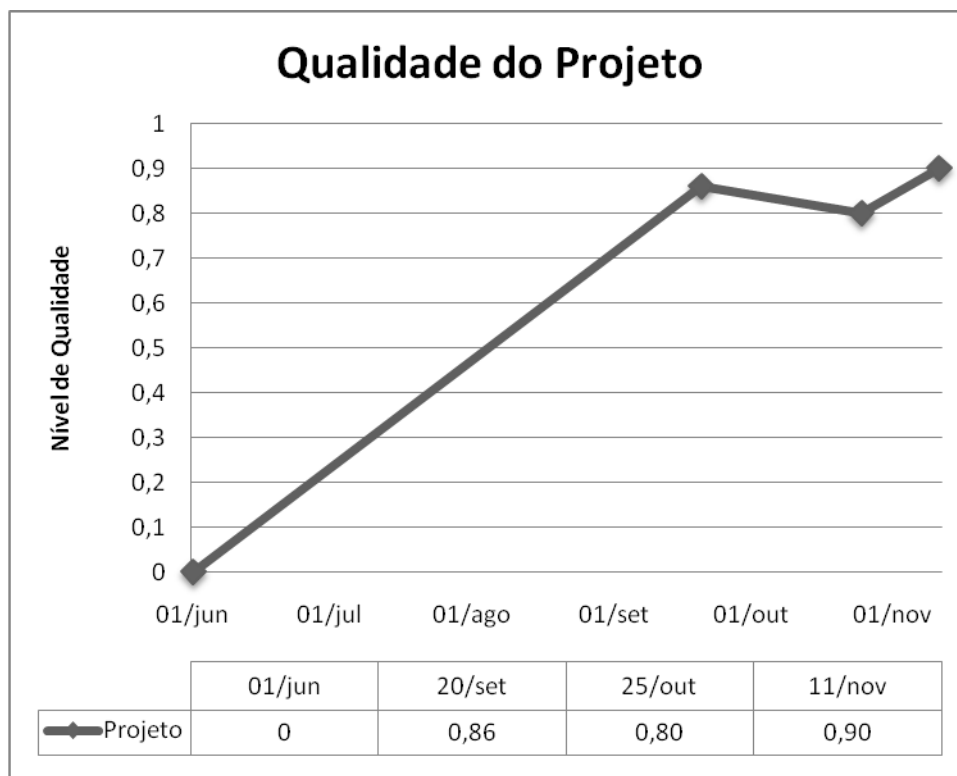


Figura 13. Qualidade do Projeto *versus* Avaliações

A primeira avaliação obteve o nível de qualidade do projeto de $\varphi_{projeto} = 0,86$, ficando abaixo do zero vírgula noventa numa escala de zero a um. As demais avaliações obtiveram um nível de qualidade de aproximadamente de zero vírgula oitenta.

O decréscimo entre a primeira e segunda avaliação ocorreu devido à inclusão de novos casos de uso no escopo de avaliação e os níveis de qualidade desses novos casos de uso influenciaram na queda do nível de qualidade do projeto, conforme dados apresentado na Subseção 4.4.1.2.2. Houve um progresso da qualidade do projeto na terceira avaliação em relação à segunda avaliação, pois houve um progresso da qualidade dos casos de uso que estava no escopo da avaliação.

4.4.1.4. Solucionar os problemas

Esta atividade tem como objetivo definir soluções para os problemas identificados durante as avaliações. Após a identificação dos problemas é preciso estabelecer diretrizes para solucionar os problemas. Entre os diversos itens, destacam-se os principais itens que levaram aos níveis de qualidade apresentados durante a aplicação do estudo de casos foram:

- O documento de requisitos não está atualizado de acordo com todos os levantamentos registrados em ata de reunião;
- Os casos de uso que estão desatualizados em relação ao documento de requisitos e alguns itens do artefato não foram devidamente preenchidos; e,
- O termo de homologação de requisitos não foi elaborado e não foi assinado pelo usuário final.

Em suma, os problemas identificados foram: alguns artefatos não são produzidos e, quando são produzidos, falta de informações relevantes que deveriam estar contidas nos artefatos. Diante do problema exposto, a solução adotada pelo gerente de projetos é estabelecer e planejar treinamentos para a equipe, visando suprir toda qualquer deficiência técnica.

Os treinamentos serão concebidos e ministrados pela equipe de qualidade, responsável, também, em manter a equipe técnica do projeto atualizada na metodologia de desenvolvimento de software e nos conceitos abordados pelo RUP (Kruchten 2003).

Posteriormente, a gerência do projeto vai incorporar diretrizes para o devido acompanhamento de todas as atividades presente na metodologia de desenvolvimento, com o objetivo de que todas as atividades e seus respectivos passos sejam seguidos e os artefatos sejam produzidos.

4.5. Considerações Finais

A realização do estudo de caso teve como objetivo avaliar o processo definido e apresentado neste trabalho, com o apoio da ferramenta Inspector Panel. A aplicação do

processo de avaliação da qualidade de software em um projeto real foi importante para identificar melhorias no processo de desenvolvimento da organização.

A definição da complexidade para os artefatos e para os casos de uso durante as fases do projeto foi essencial para representar a realidade e o dinamismo das atividades diárias do projeto. Cada organização deve determinar o nível de qualidade de aceitação para cada caso de uso e para o projeto.

Durante o estudo de caso, pode-se observar que mesmo com uma metodologia de desenvolvimento de software madura e um processo definido, com atividades estabelecidas e objetivos claros, artefatos de entrada e saída devidamente mapeados, responsáveis e ferramentas identificados, não garantem a execução das atividades da metodologia de desenvolvimento na organização. Assim sendo, precisa-se trabalhar a cultura organizacional no cumprimento efetivo e padronizado das atividades, visando garantir a qualidade dos produtos e dos serviços.

Este capítulo apresentou o uso de avaliações de progresso em projetos de software em uma organização pública do estado de Pernambuco, visando avaliar a adoção e receptividade do processo frente a um processo definido e estabelecido. Neste contexto, o cenário do projeto e o perfil da equipe foram mencionados, seguidos das três avaliações de progresso de qualidade, onde é possível observar o nível de qualidade de cada caso de uso e os resultados obtidos. O próximo capítulo apresenta as conclusões deste trabalho de dissertação, mencionado as principais contribuições do trabalho, as propostas de trabalhos futuros, e por fim, um comparativo com os trabalhos relacionados é apresentado.

5. CONCLUSÕES

*Não confunda jamais conhecimento com sabedoria.
Um o ajuda a ganhar a vida; o outro a construir uma vida.
—SANDRA CAREY*

Neste capítulo são apresentadas as principais conclusões do trabalho, enfatizando, as contribuições desta dissertação. Em seguida, as limitações do trabalho são apresentadas. Por fim, as perspectivas de trabalhos futuros e considerações finais são apresentadas.

5.1. Contribuições

Neste trabalho foram discutidos e apresentados os principais conceitos de qualidade (Crosby, 1979; Deming, 1982; Juran, 1988), abordando os principais modelos e normas nacionais e internacionais (ABNT, 1997; ABNT, 2003; ISO/IEC, 2003; SEI, 2002; SEI, 2006; Softex, 2009), visando auxiliar o gerente de projeto e a equipe de qualidade em suas atividades, de modo a estabelecer indicadores, diretrizes e ações que garantam o sucesso da qualidade do projeto de software.

Apreendeu-se que os projetos de desenvolvimento de software devem ser conduzidos através de processos padronizados e que a qualidade é um fator crítico de sucesso, devendo ser acompanhada durante todo o ciclo de vida. Avaliar a qualidade de um projeto requer conhecimentos específicos dos membros da equipe de desenvolvimento. As avaliações de qualidade realizadas tendem a ajudar a corrigir falhas do processo de desenvolvimento. A qualidade do processo influencia na qualidade do produto, que por sua vez, influencia na qualidade em uso (ABNT 2003).

Propôs-se um processo de avaliação de progresso da qualidade de software, através do mapeamento da **área de qualidade** (ISO/IEC 1994) e das características apresentadas por (Meneses 2001). As atividades da avaliação foram baseadas na ISO/IEC (1996). Definiu-se um conjunto de métricas para avaliar o progresso da qualidade que fornecem informações ao gerente sobre o progresso da qualidade dos casos de uso e do projeto como um todo, no nível de iterações e fases. Todo o processo de avaliação de progresso da qualidade foi mapeado

utilizando o *Eclipse Process Framework* (EPF), onde as atividades, os papéis foram definidos e os artefatos foram elaborados.

O presente trabalho teve as seguintes contribuições:

- Proposta de um processo flexível para avaliar o progresso da qualidade em projetos de software, baseado em modelos e normas de qualidade;
- Possibilidade de incorporar a visão de qualidade ao processo Inspector (Meneses 2001), tornando-o mais completo e amplo;
- Apoio de uma ferramenta web, Inspector Panel, para facilitar sua utilização num ambiente distribuído;
- Automatização na coleta e processamento dos dados de qualidade, definindo o nível de qualidade dos casos de uso e do projeto;
- Evitar falhas e erros durante a coleta e processamento dos dados, pois o Inspector Panel calcula as métricas estabelecidas;
- Criação de base histórica para a organização das avaliações realizadas; e,
- Geração de indicadores visuais (gráficos e tabela).

Para avaliar o processo definido, realizou-se um estudo de caso em um projeto de desenvolvimento de software em uma organização pública do estado de Pernambuco. Os resultados deste estudo de caso apresentaram indicadores sobre o progresso da qualidade do projeto. Durante o estudo de caso, observou-se que mesmo com uma metodologia de desenvolvimento de software definida e a capacidade técnica dos colaboradores atestada, algumas atividades não são executadas e alguns artefatos não são produzidos pelos seguintes motivos:

- O primeiro motivo deve ao fato de que é preciso atender aos prazos definidos pelo cliente. Alguns destes prazos são determinados em juízo, devendo ser cumpridos conforme o prazo estipulado em lei;

- O segundo motivo trata-se da falta de uma cultura organizacional em gerar todos os artefatos definidos pela metodologia de desenvolvimento. A falta de conhecimento da equipe de análise e de desenvolvimento sobre a metodologia e seu fluxo de atividades foi facilmente observada durante as avaliações de progresso;
- O terceiro motivo se dá pela falta de cobrança da gerência com relação aos artefatos, apesar da gerência apoiar a metodologia e defender o seu uso.

É notório que as mudanças estão acontecendo dentro da secretaria, onde foi aplicado o estudo de caso, visando o uso efetivo da metodologia de desenvolvimento de software. As principais mudanças são: melhor planejamento e agrupamento dos objetivos a serem atendidos e inclusão do esforço necessário para manter a documentação dos casos de uso atualizados. Certamente, estas mudanças estão sendo incorporadas na medida certa para que não haja resistência dos colaboradores em sua maioria, servidores públicos.

Freitas (2006) apresenta aspectos motivacionais que influenciam os atores no processo de desenvolvimento de software. Baseado em literatura especializada (psicologia), o autor, apresenta uma análise da pesquisa realizada com cerca de oitenta profissionais de diversas regiões do Brasil. Os aspectos relacionados ao trabalho em equipe, sistema de recompensa e punições e desenvolvimento de carreira, podem servir de diretrizes para gerar motivações nos colaboradores, seja em organização pública ou privada, facilitando mudanças significativas dentro de curto intervalo de tempo.

5.2. Comparativo entre os Trabalhos Relacionados

De modo a facilitar o entendimento de trabalhos relacionados em comparação com o trabalho apresentado nesta dissertação, a Tabela 12, apresenta os pontos mais e os pontos menos relevantes.

Tabela 12 – Comparação entre as Abordagens

Trabalho	Pontos menos Relevantes	Pontos mais Relevantes
Silva 1995	<ul style="list-style-type: none"> - Focaliza apenas a fase de testes baseada no fluxo de eventos. - Utilizado para linguagens de programação estruturadas. 	<ul style="list-style-type: none"> - Ambiente configurável. - Apoiado por uma ferramenta (PROTESTE+).
Rêgo 1996	<ul style="list-style-type: none"> - Focaliza em produtos de software prontos (prateleiras). - Utiliza apenas a visão do usuário. 	<ul style="list-style-type: none"> - Processo de avaliação da ISO/IEC 14598. - Baseado na NBR ISO/IEC 9126 e ISO/IEC 12119. - Avaliações realizadas em mais de 480 produtos de software.
Belchior 1997	<ul style="list-style-type: none"> - Requer conhecimento matemático elevado. - Uso manual exige um enorme esforço. 	<ul style="list-style-type: none"> - Processo de avaliação baseada na ISO/IEC 14598. - Utiliza o nível de experiência de especialistas. - Uso de questionários padronizados. - Apoiado por uma ferramenta (AdeQuaS).
Oliveira 2002	<ul style="list-style-type: none"> - Ambiente não distribuído. 	<ul style="list-style-type: none"> - Agilidade na execução de avaliação da qualidade. - Adaptação ao padrão NBR ISO/IEC 9126.
Marzullo 2006	<ul style="list-style-type: none"> - Uso de um mediador para transformar os artefatos no formato XML. 	<ul style="list-style-type: none"> - Pode ser aplicado a diversas metodologias de desenvolvimento. - Visualização do status de cada artefato. - Apoiado por uma ferramenta (PATT).
Inspector Q	<ul style="list-style-type: none"> - Não focaliza produtos de software de prateleira. - Não apresenta questionário padrão para definição do nível de complexidade das funcionalidades. 	<ul style="list-style-type: none"> - Processo de avaliação baseada na ISO/IEC 14598. - Pondera o nível de complexidade das funcionalidades do produto de software.

- Geração de indicadores visuais.

- Apoiado por uma ferramenta
(Inspector Panel).

Pode-se observar que a maioria das abordagens relacionadas tem o apoio de ferramentas, baseados em normas nacionais e internacionais de qualidade, as abordagens visam avaliar a qualidade dos produtos de software.

O presente trabalho se destaca por ponderar o nível de complexidade (pesos) das funcionalidades (casos de uso) do produto de software. A ferramenta Inspector Panel foi concebida como um software gratuito ou *freeware*, isto é, sua utilização não implica no pagamento de licenças de uso.

5.3. Limitações

A definição de um processo não é uma tarefa trivial. Diversas barreiras devem ser superadas na implantação do processo, entre as quais se destacam: a barreira da mudança na cultura organizacional e a barreira do sentimento de “mais um processo” ou “mais trabalho” para a equipe. Destacam-se como principais limitações enfrentadas por este trabalho:

- Restrições de tempo na realização de novos estudos de caso. Os novos estudos devem ser aplicados em empresas privadas e públicas, visando identificar pontos de melhoria no processo definido a partir de diferentes perspectivas e em contextos organizacionais diferentes;
- Baixa disponibilidade de especialistas da área de qualidade para auxiliar no processo de condução e de avaliação do processo e da ferramenta;
- Aumento de tarefas na área de qualidade em alimentar os dados na ferramenta Inspector Panel;
- Caracterizar a definição dos pesos dos artefatos para cada projeto de software na organização, dependendo da natureza do projeto e independentemente da visão de um único analista;

- Dificuldades inerentes ao desenvolvimento de software, como atrasos. Os atrasos do projeto podem adiar as avaliações de progresso da qualidade.

5.4. Trabalhos Futuros

Nesta Seção, serão apresentadas possibilidades futuras de pesquisa, seja relacionado ao processo de avaliação de progresso da qualidade ou mesmo propor novos mecanismos de acompanhamento de projetos de software. Destacam-se as seguintes possibilidades de extensão deste trabalho:

- Estender e abranger o processo de avaliação de progresso da qualidade em projeto que utilizam uma metodologia de desenvolvimento ágil. O desenvolvimento de software que utilizam metodologias ágeis vem crescendo a cada dia. Neste sentido, a possibilidade de adaptar o processo de avaliação de progresso da qualidade em projetos que utilizam um desenvolvimento ágil. O *OpenUP* (EPF 2011) e o SCRUM (Schwaber 1997) são fortes candidatos. O *OpenUP* é uma metodologia ágil de desenvolvimento de software, baseada nas principais características do RUP terá maior facilidade de adequação ao Inspector Q. O SCRUM, por sua vez, deve modificar o papel do gerente de projetos pelo papel do *Scrum Master* e o papel do coletor de informações pode ser atribuído a um ou vários membros do *Time*, dependendo do escopo da inspeção (segundo pilar do SCRUM). Cada iteração passar a ser considerado como um *Sprint*. As características do processo de avaliação de progresso da qualidade devem ser contextualizadas para um ambiente organizacional que utiliza a metodologia ágil na construção de produtos de software, respeitando suas características essenciais.
- Adaptar o processo de avaliação de progresso da qualidade do software (produto) para avaliar a qualidade do processo utilizado no desenvolvimento do software. As organizações estabelecem processos a serem seguidos por seus colaboradores, porém é preciso verificar se o processo estabelecido atende as reais necessidades da organização. Neste sentido, serão realizadas entrevistas com os colaboradores com as seguintes finalidades:

- Averiguar o uso efetivo ou não do processo no dia a dia da organização;
 - Identificar possíveis lacunas não cobertas pelo processo, sendo necessário definir ações para melhoria no processo; e,
 - Identificar atividades desnecessárias e excluí-las do processo para deixá-lo rápido e simples. O foco será a qualidade do processo, pois a qualidade do processo influencia a qualidade do produto (ABNT 2003).
- Estabelecer objetivamente diretrizes para calcular o nível de complexidade dos casos de uso e dos artefatos produzidos no projeto, gerando assim uma técnica, clara e precisa, de complexidade para casos de uso e artefatos. Atualmente, existem indicadores para definição do nível de complexidade (pesos) dos casos de uso, sendo importante padronizar para que as organizações obtenham os valores iguais de complexidade para casos de uso semelhantes.
 - Estabelecer diretrizes para o acompanhamento dos custos de cada caso de uso do projeto durante o ciclo de vida, com o objetivo de monitorar e controlar o custo real e o estimado do projeto. Sendo assim, os custos deverão ser apresentados ao nível de detalhamento da iteração do projeto de software. Com essas informações, o gerente de projetos pode acompanhar objetivamente os custos por iteração, por fases do projeto e por casos de uso.

5.5. Considerações Finais

As organizações que possuem metodologias definidas e nível de maturidade estabelecido, não garantem que seus colaboradores estejam seguindo a metodologia de desenvolvimento em sua plenitude. O não cumprimento de algumas atividades do processo pode gerar softwares defeituosos, pois o conhecimento pertencente à pessoa não está documentado, possibilitando o não entendimento de como o software deve, de fato, comportar-se.

Projetos de software em organizações públicas tendem a cumprir as regras e prazos estabelecidos em lei, sendo, se necessário, ignorar e abandonar algumas etapas do processo de desenvolvimento, bem como os padrões estabelecidos para o projeto em virtude do

cumprimento desses prazos. O sucesso do projeto, na organização, está mais direcionado aos interesses do estado, sendo assim, se for necessário alocar mais recursos para o projeto, isto poderá ser feito.

Apesar de várias propostas (Silva, 1995; Rêgo, 1996; Belchior, 1997; Oliveira 2002; Marzullo 2006) para avaliar a qualidade de software terem sido apresentadas, na academia ou na indústria, pode-se observar que mesmo levando em consideração o grau de importância das características de qualidade, não é possível identificar com clareza em que fase ou iteração do desenvolvimento de software o produto de software precisa ser melhorado.

Observa-se, também, que o grau de complexidade (peso) dos casos de uso não ganha a devida importância. A qualidade de um caso de uso complexo pode impactar, fortemente, na qualidade total do projeto. Em muitos projetos, os casos de uso complexos possuem as regras de negócios mais importantes do sistema, conseqüentemente, representam o “coração” do negócio da organização.

Este trabalho propôs um processo de avaliação de progresso da qualidade de software, através do mapeamento da área de qualidade (ISO/IEC, 1994; ISO/IEC, 1996; ABNT, 2003; Chrissis, 2003; Softex, 2009) e das características apresentadas por (Meneses 2001). As atividades do processo de avaliação foram baseadas em (ISO/IEC 1996). O processo, denominado de Inspector Q, é composto por quatro atividades desempenhadas por dois papéis.

Um conjunto de métricas para avaliação da qualidade foi definido. Tais métricas fornecem informações ao gerente do projeto de software sobre o progresso da qualidade dos casos de uso e do projeto como um todo, no nível de iterações e fases. Uma ferramenta foi desenvolvida para apoiar este processo, podendo ser aplicados em empresas públicas e privados.

Referências Bibliográficas

- (ABNT 1997) ABNT – Associação Brasileira de Normas Técnicas (2003). “NBR ISO/IEC 12207 Tecnologia de Informação – Processos de Ciclo de Vida de Software”. Rio de Janeiro, Brasil. 35 p.
- (ABNT 2003) ABNT – Associação Brasileira de Normas Técnicas (2003). “NBR ISO/IEC 9126-1 Engenharia de Software – Qualidade de Produto”. Rio de Janeiro, Brasil. 21 p.
- (Baker 2007) Baker, E.; Fisher, Matthew A.; & Goethert, Wolfhart-Basic (2007). “Basic Principles and Concepts for Achieving Quality”. Software Engineering Institute, Carnegie Mellon University.
- (Belchior 1997) Belchior, A. D. (1997) “Um Modelo *Fuzzy* para Avaliação da Qualidade de Software”. Orientadora Ana Regina C. da Rocha. Tese de Doutorado da Universidade Federal do Rio de Janeiro.
- (Bevan 1997) Bevan, N.; Azuma, M. (1997) “Quality in Use: Incorporating Human Factors into the Software Engineering Lifecycle”, Proceedings of the 3rd IEEE International Software Engineering Standards Symposium and Forum, Walnut Creek, Canada, June, pp. 169-179.
- (Borges 2006) Borges, V. R. (2006) “Implantação de Práticas de Gerência de Configuração em uma Fábrica de Software: Um Estudo de Caso”. Orientador André Luiz Zambalde. Monografia da Universidade Federal de Lavras.
- (Chrissis 2003) Chrissis, M.; Konrad, M.; Shrum, S. “CMMI: Guidelines for Process Integration and Product Improvement”. Addison-Wesley Pub Co, 2003.
- (Cooper 2002) Cooper, J. & Fisher, M., eds. (2002) “Software Acquisition Capability Maturity Model (SA-CMM)”. Version 1.03 (CMU/SEI-2002-TR-010,

- ADA399794). Software Engineering Institute, Carnegie Mellon University.
- (Crosby 1979) Crosby, P. B. “Quality is Free: The Art of Making Quality Certain”. New York, EUA: McGraw-Hill, 1979.
- (Deming 1982) Deming, W. E. “Quality, Productivity and Competitive Position”. Massachusetts Inst Technology, Center for Advanced Engineering Study, USA,1982.
- (Deming 1990) Deming, W. E. “Qualidade: a Revolução da Administração”. Rio de Janeiro, Brasil: Marques Saraiva, 1990.
- (EPF 2011) The Eclipse Foundation. OpenUp. Acesso em 14 de Junho de 2011, disponível em <http://epf.eclipse.org/wikis/openup/index.htm>
- (Freitas 2006) Freitas, S. F. (2006) “Análise de Aspectos Motivacionais que Influenciam os Atores no Processo de Software”. Orientador Arnaldo Dias Belchior. Dissertação de Mestrado, Universidade de Fortaleza.
- (Humphrey 1990) Humphrey, W. “Managing the Software Process”. Addison-Wesley, 1990. 494p.
- (IEEE 1028 1997) IEEE Std 1028–1997 (R2002), IEEE Standard for Software Reviews: IEEE, 1997.
- (ISO/IEC 1994) the International Organization for Standardization and International Electrotechnical Commission (1994) “ISO/IEC 8402 – Quality Management and Quality Assurance – Vocabulary”.
- (ISO/IEC 1996) the International Organization for Standardization and International Electrotechnical Commission (1996) “ISO/IEC 14598-1 – Software Product Evaluation – Part 1: General Overview”.

- (ISO/IEC 2003) the International Organization for Standardization and International Electrotechnical Commission (2003) “ISO/IEC 15504 Information Technology – Software Process Assessment”.
- (Juran 1988) Juran, J. M. “Juran on Planning for Quality”. New York, EUA: Free Press Collier Macmillan, 1988.
- (Kerzner 2002) Kerzner, H. “Gestão de Projetos: As Melhores Práticas”. Porto Alegre, Brasil: Bookman, 2002.
- (Kotonya 1998) Kotonya, G.; Sommerville, I. “Requirements Engineering: Processes and Techniques”. New York, EUA: John Willey & Sons Ltd, 1998.
- (Kruchten 2003) Kruchten, P. “Introdução ao RUP – Rational Unified Process”. 2. ed. Ciência Moderna, 2003. 272 p.
- (Machado 2001) Machado, L. F. D. C. et al. (2001) “Experiência na Definição e Implantação de Processos de Software”. Workshop de Qualidade de Software – WQS.
- (Magalhães 2006) Magalhães, A. L. (2006) “A Garantia da Qualidade e o SQA: Sujeito que Ajuda e Sujeito que Atrapalha”, In: II Workshop de Implementadores MPS.Br, Recife, Brasil.
- (Mangold 2007) Mangold, P. “TI: Gerenciamento de Projetos”. 1. ed. Rio de Janeiro, Brasil: Campus, 2007. 160 p.
- (Marckzak 2003) Marckzak, S. (2003) “Uma Proposta de Organização e Funcionamento da Função de Garantia de Qualidade de Software em um Contexto de Implantação do SW-CMM”, In: Anais do II Simpósio Brasileiro de Qualidade de Software, Fortaleza, Brasil.
- (Marzullo 2006) Marzullo, F. P. (2006) “Uma Abordagem para Acompanhamento de Projetos de Software”. Orientador Geraldo Bonorino Xexéo. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro.

- (Meneses 2001) Meneses, J. (2001) “Inspector: Um Processo de Avaliação de Progresso para Projetos de Software”. Orientador Hermano Perrelli de Moura. Dissertação de Mestrado, Centro de Informática da Universidade Federal de Pernambuco.
- (Monteiro 2005) Monteiro, T. C. (2005) “Pontos de Caso de Uso Técnicos (TUCP): Uma Extensão da UCP”. Orientador Arnaldo Dias Belchior. Dissertação de Mestrado, Universidade de Fortaleza.
- (Montoni 2006) Montoni, M.; Santos, G.; Figueiredo, S.; Filho, R. C.; Barcelos, R.; Barreto, A.; Cerdeiral, C.; Lupo, P.; Rocha, A. R. (2006) “Uma Abordagem de Garantia de Qualidade de Processos e Produtos de Software com Apoio de Gerência de Conhecimento na Estação TABA”, In: V Simpósio Brasileiro de Qualidade de Software, Vila Velha, Brasil.
- (Moreira 2004) Moreira, R. T. (2004) “Gestão do Conhecimento em Qualidade de Software: Construção de um Portal da Qualidade de Software para o Brasil”. Orientadora Ana Cristina Rouiller. Monografia da Universidade Federal de Lavras.
- (Nojoza 2005) Nojoza, L. A. (2005) “Gerenciamento da Qualidade: Uma Nova Disciplina para o RUP”. Orientador Arnaldo Dias Belchior. Dissertação de Mestrado da Universidade de Fortaleza.
- (Oliveira 1999) Oliveira, S. L. “Tratado de Metodologia Científica”. São Paulo, Brasil: Pioneira, 1999.
- (Oliveira 2002) Oliveira, K. (2002) “AdeQuaS: Ferramenta *Fuzzy* para Avaliação da Qualidade de Software”. Dissertação da Universidade de Fortaleza. Mestrado em Informática Aplicada, Fortaleza.
- (Oliveira 2007) Karine, C. A. (2007) “Um processo de Gerência de Configuração Baseado no nível 2 do CMMI Estagiado para Fábricas de Software Orientadas a Produto”. Orientador Alexandre Marcos Lins

- Vasconcelos. Dissertação de Mestrado, Centro de Informática da Universidade Federal de Pernambuco.
- (Paulk 1993) Paulk, M. C.; Curtis, B.; Chrissis, M. B.; Weber, C. V. “Capability Maturity Model for Software”. Software Engineering Institute, 1993.
- (PMBOK 2004) PMBOK Guide (2004). “A Guide to the Project Management Body of Knowledge”. 3. ed. Pennsylvania: PMI.
- (PMBOK 2008) PMBOK Guide (2008). “A Guide to the Project Management Body of Knowledge”. 3. ed. Pennsylvania: PMI.
- (Prado 2003) Prado, D. “Gerenciamento de Projetos nas Organizações”. 2. ed., Belo Horizonte, Brasil: Editora de Desenvolvimento Gerencial, 2003. 199 p.
- (Pressman 2006) Pressman, R. S. “Engenharia de Software”. 6. ed. São Paulo, Brasil: Mcgraw-hill, 2006. 720 p.
- (Rêgo 1996) Rêgo, C. et al. (1996) “MEDE-PROS – Método de Avaliação de Qualidade de Produtos e Software”, versão 1.0. Campinas. Patente junto à Fundação Biblioteca Nacional sob número de registro 135.620, livro 216, folha 84. Pedido de registro de marca junto ao INPI sob número 820166243.
- (Rocha 1993) Rocha, A. R. C.; Campos, G. H. B. (1993) “Avaliação da Qualidade de Software Educacional”, In: Em Aberto.
- (Rocha 2001) Rocha, A. R. C. et al.. “Qualidade de software: Teoria e prática”. Prentice Hall, 2001.
- (Schwaber 1997) Schwaber, K., Scrum Development Process, in OOPSLA Business Object Design and Implementation Workshop, J. Sutherland, et al., Editors. 1997, Springer: London.
- (SEI 2002) Software Engineering Institute. (2002) “CMMI-SW for Systems Engineering/Software Engineering” .Version 1.1 - CMU/SEI-2002-TR-

012. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. Disponível em: <<http://www.sei.cmu.edu/library/abstracts/reports/02tr002.cfm>>. Acessado em 05/09/2010.
- (SEI 2006) Software Engineering Institute. (2006) “CMMI for Development (CMMI-DEV)”. Version 1.2, Technical Report CMU/SEI-2006-TR-008. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. Disponível em: <<http://www.sei.cmu.edu/library/abstracts/reports/06tr008.cfm>>. Acessado em 05/09/2010.
- (Silva 1995) Silva, J. B. (1995) “PROTESTE+: Ambiente de Validação Automática de Qualidade de Software Através de Técnicas de Teste e de Métricas de Complexidade”. Orientadora Ana Maria de Alencar Price. Dissertação de Mestrado da Universidade Federal do Rio Grande do Sul.
- (Simão 2002) Simão, R. P. S. (2002) “Características de Qualidade para Componentes de Software”. Orientador Arnaldo Dias Belchior. Dissertação de Mestrado da Universidade de Fortaleza.
- (Softex 2009) Softex – Sociedade para Promoção da Excelência do Software Brasileiro (2009) “MPS.BR – Melhoria de Processo do Software Brasileiro, Guia Geral”.
- (Spinelli 2005) Spinelli, A. C. P. (2005) “ENVISION: Um Modelo para Avaliação de Produtos de Software Baseado em Visões Extensíveis”. Orientador Hermano Perrelli de Moura. Monografia em Qualidade de Software, Centro de Informática da Universidade Federal de Pernambuco.
- (Unhelkar 2003) Unhelkar, B. “Process Quality Assurance for UML-based projects”. In: The Addison-Wesley, 2003.

- (Vicente 2008) Vicente, A. (2008) “Inspector Panel: Um Painel de Controle para Acompanhamento de Progresso de Projetos de Software”. Orientador Hermano Perrelli de Moura. Monografia da Especialização em Gestão de Tecnologia da Informação, Centro de Informática da Universidade Federal de Pernambuco.
- (Villas-Boas 2005) Villas-Boas, A.; Côrtes, M. L.; Lopes, S. (2005) “Do SQA ao PPQA: a Implantação do CMMI do CPqD”, In: VII Simpósio Internacional de Melhorias de Processos de Software, São Paulo, Brasil.
- (Wheeler 1998) Wheeler, S.; Duggins, S. (1998) “Improving Software Quality”, Proceedings of the 36th Annual Conference on Southeast Regional Conference, April, pp. 300-309.
- (Yin 2001) Yin, R. K. “Estudo de caso – planejamento e métodos”. (2Ed.). Porto Alegre, Brasil: Bookman, 2001.

Apêndices

A seguir são apresentados os templates para artefatos usados pelo processo, conforme abaixo:

- Apêndice A: Critérios de Avaliação da Qualidade dos Artefatos
- Apêndice B: Checklist de Avaliação da Qualidade dos Artefatos



Critérios de Avaliação da Qualidade dos Artefatos

Projeto *<nome_projeto>*

<Esta página se refere ao template deste documento. Deve-se excluir no momento da elaboração deste documento.>

Histórico de Alterações do Template

Data	Versão	Responsável	Descrição
04/03/2010	01.00	Aristides Vicente	Criação do documento

Lista de Aprovadores do Template

Nome	Cargo	Data da Aprovação

Histórico de Alterações

<Registre as alterações feitas neste documento.>

Data	Versão	Responsável	Descrição

Lista de Aprovadores

<Registre aqui os responsáveis por aprovar este documento.>

Nome	Cargo	Data da Aprovação

Índice

1. IDENTIFICAÇÃO	107
2. CRITÉRIOS DE QUALIDADE	107
3. REFERÊNCIAS	107

1. Identificação

Nome do Projeto: <Descreve o nome do projeto>

Descrição do Projeto: <Descreve sucintamente a ideia do projeto>

Responsável pelo Documento: <Descreve o responsável pelo preenchimento do documento>

2. Critérios de Qualidade

<Determina os artefatos que poderão ser produzidos durante o desenvolvimento do projeto e identifica quais os critérios para avaliar se os mesmos contêm as informações de um determinado caso de uso. Alguns artefatos padrões já estão inseridos na guia de Critérios para Inspeção dos Artefatos. Se o artefato inserido na guia for produzido nesse projeto, basta copiar o item relacionado para esse documento (ou fazer referência para a guia), caso contrário, os critérios deverão ser definidos e documentados, além de armazenados e incluídos na guia como forma de futura referência.>

2.1. <Descrição do critério>

<Lista os critérios para avaliar se o artefato foi alterado ou criado para conter as informações relativas à realização das etapas de desenvolvimento de um determinado caso de uso.>

2.2. <Descrição do critério>

2.3. <Descrição do critério>

3. Referências

<Qualquer documento que foi utilizado como fonte de informação para produção do artefato Critério de Avaliação da Qualidade dos Artefatos.>

- Plano de Projeto
- Plano de Qualidade

Anexos

A seguir são apresentados os templates para artefatos usados pelo processo, conforme abaixo:

- Anexo A: Plano de Avaliação do Progresso
- Anexo B: Formulário para Coleta de Informações de Progresso sobre Caso de Uso
- Anexo C: Formulário para Coleta de Informações de Progresso sobre o Projeto
- Anexo D: Relatório de Avaliação do Progresso de Qualidade



Plano de Avaliação do Progresso

Projeto *<nome_projeto>*

<Esta página se refere ao template deste documento. Deve-se excluir no momento da elaboração deste documento.>

Histórico de Alterações do Template

Data	Versão	Responsável	Descrição
02/03/2010	01.00	Aristides Vicente	Revisão e refinamento do documento
11/03/2010	01.01	Aristides Vicente	Inserção de referências a ferramenta Inspector Panel
19/07/2010	01.02	Aristides Vicente	Alteração do nome do <i>template</i> de PlanoAvaliacaoProgressoTecnico.docx para PlanoAvaliacaoProgresso.docx

Lista de Aprovadores do Template

Nome	Cargo	Data da Aprovação

Histórico de Alterações

<Registre as alterações feitas neste documento.>

Data	Versão	Responsável	Descrição

Lista de Aprovadores

<Registre aqui os responsáveis por aprovar este documento.>

Nome	Cargo	Data da Aprovação

Índice

1. IDENTIFICAÇÃO	114
2. ESCOPO	114
3. METAS DA AVALIAÇÃO	114
4. DATAS E RESPONSABILIDADES	114
5. REFERÊNCIAS	115

1. Identificação

Nome da Organização: <Descreve o nome organização>

Área de Atuação: <Descreve a área de atuação da organização>

Localização: <Descreve a localização da organização>

2. Escopo

<Determina o escopo da avaliação de progresso a ser realizada. Identifica os casos de uso e atividades focalizados nessa avaliação, bem como o intervalo de tempo de realização do projeto que será considerado.>

<Antes da inspeção, a ferramenta **Inspector Panel** disponibiliza uma lista de casos de uso onde é possível determinar o escopo da inspeção. Ou seja, define os casos de uso que irão fazer parte da inspeção.>

3. Metas da Avaliação

<Apresenta quais são os resultados esperados durante a avaliação de progresso. Tais metas são definidas a partir das informações contidas em artefatos de apoio (Plano de Projeto, Plano de Iteração), da análise de tendências da avaliação anterior, além de experiências anteriores com projetos semelhantes.>

4. Datas e Responsabilidades

<Essa Seção consiste em uma tabela, onde cada linha define a equipe a ser avaliada, o coletor (responsável pela coleta dos dados relativos a uma equipe), os casos de uso que ele deverá coletar as informações de progresso e a data que a coleta deverá ser realizada. Tais informações serão usadas no preenchimento do artefato Formulário para Coleta de Informações sobre o Progresso sobre Caso de Uso.>

<A ferramenta **Inspector Panel** realiza a inspeção e expõe os casos de uso que serão focalizados. A data da coleta é a data/hora do sistema e todas as equipes responsáveis pelo desenvolvimento dos casos de uso, selecionados, são avaliados.>

Tabela 13 – Responsáveis pela Coletas de Dados

Coletor	Equipe a ser avaliada	Casos de Uso Focalizados	Data da Coleta
<Nome do coletor de informações>	<Nome da equipe que está sendo avaliada>	<Casos de uso que estão sendo avaliados>	<dd/mm/aaa>

5. Referências

<Qualquer documento que foi utilizado como fonte de informação para produção do Plano de Avaliação do Progresso.>

- Plano de Projeto
- Plano de Iteração

Anexo B – Template: Formulário para Coleta de Informações de Progresso sobre Caso de Uso



**Formulário para Coleta de Informação
de Progresso sobre Caso de Uso**

Projeto *<nome_projeto>*

<Esta página se refere ao template deste documento. Deve-se excluir no momento da elaboração deste documento.>

Histórico de Alterações do Template

Data	Versão	Responsável	Descrição
04/03/2010	01.00	Aristides Vicente	Revisão e refinamento do documento
19/07/2010	01.01	Aristides Vicente	Inserção dos aspectos de qualidade Inclusão de documentos na seção de referências

Lista de Aprovadores do Template

Nome	Cargo	Data da Aprovação

Histórico de Alterações

<Registre as alterações feitas neste documento.>

Data	Versão	Responsável	Descrição

Lista de Aprovadores

<Registre aqui os responsáveis por aprovar este documento.>

Nome	Cargo	Data da Aprovação

Índice

1. IDENTIFICAÇÃO	120
2. FASE: <NOME_FASE>	120
3. FASE: <NOME_FASE>	122
4. FASE: <NOME_FASE>	122
5. FASE: <NOME_FASE>	123
6. COMENTÁRIOS	124
7. REFERÊNCIAS	124

1. Identificação

Nome do Projeto: <Descreve o nome do projeto>

Identificação do Caso de Uso: <Identificação do caso de uso>

Nome do Caso de Uso: <Nome do caso de uso>

Descrição do Caso de Uso: <Descreve sucintamente sobre o caso de uso>

Responsável pelo Caso de Uso: <Descreve o responsável pelo caso de uso>

Responsável pela Coleta: <Descreve o responsável pelos dados da coleta>

Data de Avaliação: <dd/mm/aaaa>

Estado Anterior: <Informar o estado anterior do caso de uso, ou seja, o estado da última inspeção.>
(Caso seja a primeira vez, excluir este item)

Data de Avaliação Anterior: <dd/mm/aaaa> (Caso seja a primeira vez, excluir este item)

2. Fase: <nome_fase>

- **Documentação Inicial**

Documentação Inicial: CONCLUÍDO NÃO CONCLUÍDO

$\varphi_{documentação\ inicial} = \frac{\text{<qualidade do artefato>}}{\text{_____}}$

- **Descrição Funcional + Fluxo de Eventos**

Descrição Funcional

	CONCLUÍDO	NÃO CONCLUÍDO	NÃO NECESSÁRIO
Descrição do Caso de Uso	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entradas Definidas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Saídas Definidas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pré-Requisitos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pós-Requisitos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

$\varphi_{descricao\ funcional} =$ _____

Fluxo de Eventos

	CONCLUÍDO	NÃO CONCLUÍDO	NÃO NECESSÁRIO
Fluxo Principal	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fluxo Alternativo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subfluxos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

$\varphi_{fluxo\ de\ eventos} =$ _____

- **Interface Prototipada**

	CONCLUÍDO	NÃO CONCLUÍDO	NÃO NECESSÁRIO
Especificação da Interface	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Projeto de Interface	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementação do Protótipo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

$\varphi_{interface} =$ _____

Realização da Especificação:	TOTAL <input type="checkbox"/>	PARCIAL <input type="checkbox"/>	NENHUMA <input type="checkbox"/>
$\varphi_{especificacao} =$ _____			

3. Fase: <nome_fase>

<u>Diagramas UML:</u>	REFLETE O CASO DE USO	NÃO REFLETE O CASO DE USO	NÃO NECESSÁRIO
Seqüência/Colaboração	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Classe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estado	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Atividade	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$\varphi_{seqüencia/colaboração} =$	_____		
$\varphi_{classe} =$	_____		
$\varphi_{estado} =$	_____		
$\varphi_{atividade} =$	_____		

Realização da Análise + Projeto:	TOTAL	PARCIAL	NENHUMA
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$\varphi_{ana+prj} =$	_____		

4. Fase: <nome_fase>

	CONCLUÍDO	NÃO CONCLUÍDO	NÃO NECESSÁRIO
Diagrama de Componentes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementação das Classes Relacionadas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interface com Funcionalidades Disponíveis	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Testes de Unidade das Classes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$\varphi_{diagramadecomponentes} =$	_____		
$\varphi_{implementacao classe} =$	_____		

$\varphi_{interface} =$ _____

$\varphi_{atividade} =$ _____

Realização da Implementação:	TOTAL <input type="checkbox"/>	PARCIAL <input type="checkbox"/>	NENHUMA <input type="checkbox"/>
$\varphi_{implementacao} =$ _____			

5. Fase: <nome_fase>

	CONCLUÍDO	NÃO CONCLUÍDO	NÃO NECESSÁRIO
Plano de Teste do Caso de Uso	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Projeto de Teste do Caso de Uso	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementação do Teste do Caso de Uso	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

% dos testes do caso de uso realizados com sucesso = _____

$\varphi_{planode teste} =$ _____

$\varphi_{projetode teste} =$ _____

$\varphi_{imp. do teste} =$ _____

Realização dos Testes:	TOTAL <input type="checkbox"/>	PARCIAL <input type="checkbox"/>	NENHUMA <input type="checkbox"/>
$\varphi_{teste} =$ _____			

	TOTAL	PARCIAL	NENHUMA
Realização do Caso de Uso:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fases Realizadas:	<input type="checkbox"/>		
Especificação	<input type="checkbox"/>		
Análise + Projeto	<input type="checkbox"/>		
Implementação	<input type="checkbox"/>		
Teste	<input type="checkbox"/>		
Qualidade do Caso de Uso ($\varphi_{casodeuso}$):	_____		
Incremento Percentual ($\mathcal{P}_{casodeuso}$):	_____		

6. Comentários

<Para cada problema deverá ser criada uma Subseção descrevendo o problema, identificando possíveis soluções e escolhendo a solução mais adequada.>

7. Referências

<Qualquer documento que foi utilizado como fonte de informação, para produção do artefato Formulário de Coleta de Informação de Progresso sobre o Caso de Uso.>

- Plano de Avaliação do Progresso
- Checklist de Avaliação da Qualidade dos Artefatos

Anexo C – Template: Formulário para Coleta de Informação de Progresso sobre o Projeto



Formulário para Coleta de Informação de Progresso sobre o Projeto

Projeto *<nome_projeto>*

<Esta página se refere ao template deste documento. Deve-se excluir no momento da elaboração deste documento.>

Histórico de Alterações do Template

Data	Versão	Responsável	Descrição
02/03/2010	01.00	Aristides Vicente	Revisão e refinamento do documento
19/07/2010	01.01	Aristides Vicente	Inserção de aspectos de qualidade Inclusão de documentos na seção de referências

Lista de Aprovadores do Template

Nome	Cargo	Data da Aprovação

Histórico de Alterações

<Registre as alterações feitas neste documento.>

Data	Versão	Responsável	Descrição

Lista de Aprovadores

<Registre aqui os responsáveis por aprovar este documento.>

Nome	Cargo	Data da Aprovação

Índice

1. IDENTIFICAÇÃO	129
2. ESTADO DOS CASOS DE USO DO PROJETO	129
3. ESTADO DO PROJETO.....	130
4. COMENTÁRIOS	130
5. REFERÊNCIAS	130

1. Identificação

Nome da Organização: <Descreve o nome organização>

Tipo de Aplicação: <Descreve o tipo de aplicação do projeto>

Nome do Projeto: <Nome do projeto>

Gerente do Projeto: <Nome do gerente responsável pelo projeto>

Breve Descrição do Projeto: <Apresenta uma breve descrição do projeto>

Data de Avaliação: <dd/mm/aaaa>

Estado Anterior: <Informar o estado anterior do caso de uso, ou seja, o estado da última inspeção.>
(Caso seja a primeira vez, excluir este item)

Data Avaliação Anterior: <dd/mm/aaaa>

2. Estado dos Casos de Uso do Projeto

<Apresenta uma introdução ao documento, descrevendo rapidamente características gerais da organização, tais como: número de empregados, volume de negócios, tempo de existência, negócios realizados, etc.>

<Nesta Seção é identificado os casos de uso do projeto, suas prioridades, a equipe responsável por cada caso de uso e a qualidade obtida durante a avaliação de progresso da qualidade.>

Tabela 14 – Identificação dos Casos de Uso

Identificação do Caso de Uso	Prioridade do Caso de Uso	Equipe Responsável	$\varphi_{\text{caso de uso}}$
<UC_001>	<Alta, assume valor 10 (por exemplo)>	<Equipe de desenvolvimento>	<Progresso da qualidade do caso de uso>

3. Estado do Projeto

<Descreve os principais objetivos da avaliação do status da organização, indicando os fatores que foram observados durante o período de avaliação da organização e para que este artefato está sendo produzido.>

Qualidade do Projeto ($\varphi_{projeto}$):

Incremento Percentual ($\varphi_{projeto}$):

Resultado Esperado: <Sim ou Não>

4. Comentários

<Documenta a situação do processo de desenvolvimento da organização e o resultado da avaliação da utilização de métricas na organização, focalizando aspectos de progresso.>

5. Referências

<Qualquer documento que foi utilizado como fonte de informação para produção do artefato Modelo de Coleta de Informações de Progresso sobre o Projeto.>

- Plano de Avaliação do Progresso
- Formulário para Coleta de Informações de Progresso sobre o Caso de Uso



Relatório de Avaliação do Progresso de Qualidade

***<Primeira>* Avaliação**

Projeto *<nome_projeto>*

<Esta página se refere ao template deste documento. Deve-se excluir no momento da elaboração deste documento.>

Histórico de Alterações do Template

Data	Versão	Responsável	Descrição
04/03/2010	01.00	Aristides Vicente	Criação do documento
11/03/2010	01.01	Aristides Vicente	Inserções de referência a ferramenta Inspector Panel

Lista de Aprovadores do Template

Nome	Cargo	Data da Aprovação

Histórico de Alterações

<Registre as alterações feitas neste documento.>

Data	Versão	Responsável	Descrição

Lista de Aprovadores

<Registre aqui os responsáveis por aprovar este documento.>

Nome	Cargo	Data da Aprovação

Índice

1. IDENTIFICAÇÃO	135
2. INTRODUÇÃO	135
3. PROGRESSO	135
4. PROBLEMAS IDENTIFICADOS	138
5. CONCLUSÕES	138
6. REFERÊNCIAS	139

1. Identificação

Nome do Projeto: <Descreve o nome do projeto.>

Descrição do Projeto: <Descreve sucintamente a ideia do projeto.>

Responsável pela Avaliação: <Descreve o responsável pela avaliação do projeto junto a organização.>

Data de Avaliação: <dd/mm/aaaa>

2. Introdução

<Faz uma introdução ao documento de avaliação, identificando porque ele é produzido e qual sua importância. Além disso, informa o momento da avaliação de progresso, fazendo referência ao escopo e às metas definidas no Plano de Avaliação do Progresso.>

3. Progresso

<Essa Seção preocupa-se em documentar o progresso da qualidade dos casos de uso pertencente ao sistema, devem ser salientadas variações inesperadas no progresso da qualidade, tanto positivamente quanto negativamente.>

10.1. Tabela de Progresso

<Resume em uma tabela as informações relativas ao progresso nas funcionalidades de cada caso de uso, bem como as informações do progresso funcional do projeto. Considera também o progresso de cada etapa para o desenvolvimento do caso de uso e do projeto. A tabela abaixo consiste em um exemplo de progresso da qualidade por fases do projeto.>

<A ferramenta **Inspector Panel** apresenta uma tabela resumo indicando o percentual de progresso obtido durante as fases de desenvolvimento de software, bem como apresenta o progresso total do projeto.>

Tabela 15 – Progresso da Qualidade

Caso de Uso	Fase 1	Fase 2	Fase 3	Fase 4	Progresso Total

<UC_01>	0,66	0,66	0,00	0,00	0,33
<UC_02>	1,00	1,00	0,66	0,50	0,79
<UC_03>	0,75	0,66	0,00	0,00	0,35
<UC_04>	0,75	0,66	0,00	0,00	0,35
<UC_05>	1,00	1,00	1,00	1,00	1,00
Projeto	0,94	0,92	0,64	0,58	0,77

<Deverá conter a documentação e discussão dos dados mostrados na Tabela 15 – Progresso da Qualidade, identificando as principais variações de progresso observadas em relação à tabela gerada na avaliação anterior (se houver). Aqui, poderá ser criada uma tabela auxiliar, contendo informações sobre o incremento de funcionalidade obtido por cada caso de uso, e pelo projeto, desde a última avaliação.>

10.2. Gráficos de Progresso

<Deverá conter uma Subseção para o projeto e uma para cada caso de uso pertencente ao projeto.>

10.2.1. Progresso do <Nome do Projeto>

<Deverá ser inserido o gráfico de linha descrevendo o progresso do projeto em relação ao tempo, poderão ser realizadas análise de tendências sobre o mesmo. Abaixo segue um exemplo de gráfico de linha com informações sobre o progresso do projeto. >

<A ferramenta **Inspector Panel** gera o(s) gráfico(s) do progresso do projeto inspecionado em relação quantidade de inspeções realizadas. >

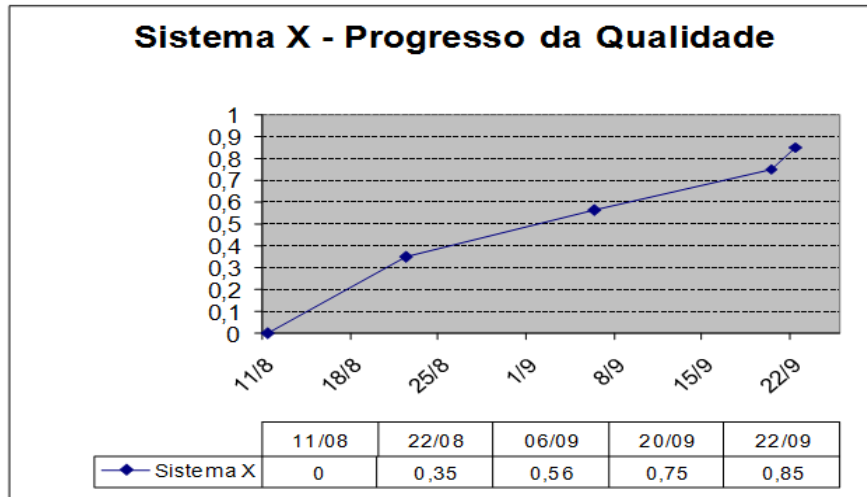


Figura 14. Exemplo do Progresso da Qualidade do Projeto

<Documentação da análise do gráfico representado. Deverão ser descritos possíveis problemas identificados através da observação de progresso e/ou tendência não esperado.>

10.2.2. Progresso do <Nome do Caso de Uso>

<Gráfico de linha descrevendo o progresso do caso de uso em relação ao tempo, realizando análise de tendências sobre o mesmo. Abaixo segue um exmplo de gráfico de linha com informações sobre o progresso de um caso de uso.>

<A ferramenta **Inspector Panel** gera o gráfico do progresso dos casos de uso do projeto inspecionado em relação quantidade de inspeções realizadas. >

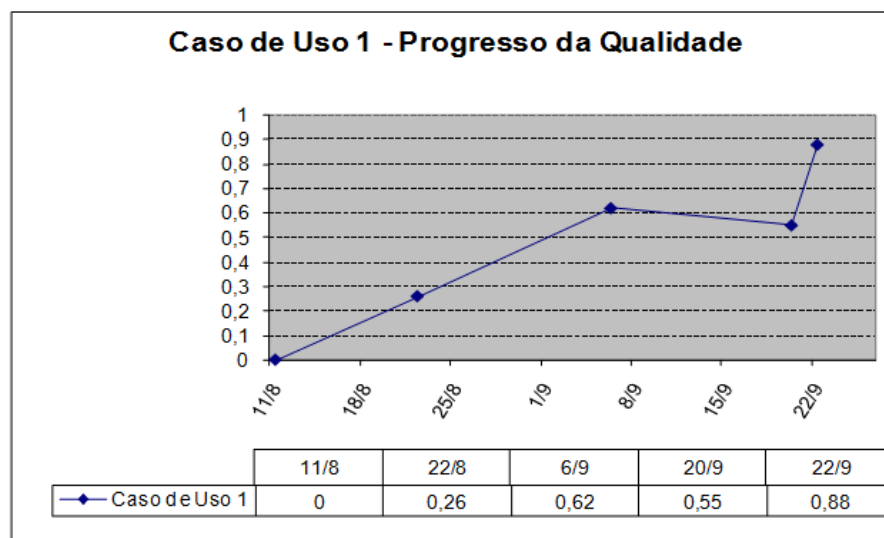


Figura 15. Exemplo do Progresso da Qualidade de Caso de Uso

<Documentação da análise do gráfico representado. Deverão ser descritas as grandes variações de valores de progresso identificados no desenvolvimento do caso de uso, pois esse tipo de variação indica que algum problema possivelmente está acontecendo na realização do caso de uso.>

10.3. Fatores Internos

<Apresenta o ambiente de desenvolvimento em termos da comunicação do pessoal, local de trabalho, recursos existentes para suporte ao desenvolvimento, segurança e demais fatores organizacionais que podem influenciar no desenvolvimento de um sistema.>

10.4. Fatores Externos

<Identifica o perfil dos clientes da organização, os competidores, o que se pode com aprender deles, os outros *stakeholders*, fornecedores, parceiros, etc. Ou seja, apresenta todas as variáveis externas que influenciam no desenvolvimento dentro da organização.>

4. Problemas Identificados

<Essa Seção contém um resumo de todos os problemas e riscos problemáticos identificados durante a avaliação do progresso técnico do projeto. Estes problemas serão colocados de forma breve em uma tabela, de modo a facilitar a leitura e entendimento dos memos.>

Tabela 16 – Problemas Identificados

ID	Problema	Descrição do Problema
<01>	<Problema X>	<O problema X>

5. Conclusões

<Documentação das conclusões retiradas a partir dessa avaliação, listando as dificuldades enfrentadas durante a recuperação das informações necessárias para avaliação de progresso. Visão geral dos principais problemas encontrados, e avaliação inicial do impacto que esses problemas podem causar se não forem tratados.>

6. Referências

<Qualquer documento que foi utilizado como fonte de informação, para produção do artefato Avaliação do Progresso.>

- Plano de Avaliação do Progresso