

UNIVERSIDADE DE CAXIAS DO SUL

ALDENARA MOREIRA SILVA

UM PLANO DE GARANTIA DA QUALIDADE PARA O *SOFTWARE* PROMOB

CAXIAS DO SUL, RS

2012

UNIVERSIDADE DE CAXIAS DO SUL
Centro de Computação e Tecnologia da Informação
Curso de Bacharelado em Sistemas de Informação

ALDENARA MOREIRA SILVA

UM PLANO DE GARANTIA DA QUALIDADE PARA O SOFTWARE PROMOB

Trabalho de Conclusão de Curso para obtenção
do Grau de Bacharel em Sistemas de
Informação pela Universidade de Caxias do Sul.
Orientador: Profa. Msc. Iraci Cristina da Silveira
De Carli

CAXIAS DO SUL, RS

2012

RESUMO

A Qualidade de *Software* possui duas dimensões: processo e produto. Normas e modelos de processos definem o processo chamado de Garantia da Qualidade, que estabelece que a qualidade deve estar presente em todas as etapas de desenvolvimento, possibilitando identificar falhas desde as primeiras etapas do processo, reduzindo custos e o impacto de um defeito, além de assegurar a aderência dos processos e produtos de trabalho aos padrões e estabelecidos pela organização. Resultante deste processo, o Plano de Garantia da Qualidade é o documento que estabelece as qualidades desejadas para o produto e como essas qualidades devem ser avaliadas. Para idealização do trabalho, uma proposta específica de um plano de Garantia da qualidade foi elaborado para aplicação em um estudo de caso.

Palavras-chaves: plano de garantia da qualidade, característica de qualidade, verificação, validação.

ABSTRACT

The *Software Quality* it is a two dimension concern: process and product. Standards and process models define the process called Quality Assurance, which establishes that the quality must be present at all stages of development, identifying possible failures from the earliest stages of the process, reducing costs and the impact of a defect, as well as ensuring the compliance of processes and work products along the standards established by the organization. Resulting from this process, the Quality Assurance Plan is the document that establishes the desired qualities for the product and how those qualities should be evaluated. For idealization of the work, a specific proposal for a quality assurance plan was developed for application study case.

Keywords: quality assurance plan, quality characteristic, verification, validation.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – Modelo de qualidade externa e interna	6
FIGURA 2 – Modelo de qualidade em uso	7
FIGURA 3 – Processos ISO/IEC 12207.....	10
FIGURA 4 – CMMI nível 2	12
FIGURA 5 – CMMI nível 3	13
FIGURA 6 – MPS.BR nível F.....	15
FIGURA 7 – MPS.BR nível D.....	16
FIGURA 8 – Etapas do planejamento de qualidade	19
FIGURA 9 – Processo de garantia da qualidade de <i>software</i>	20
FIGURA 10 – Etapas da garantia da qualidade.....	20
FIGURA 11 – Padrão de revisão de <i>software</i>	21
FIGURA 12 – Pilares da qualidade de <i>software</i>	27
FIGURA 13 – Organograma da Promob	32
FIGURA 14 – Tipos de entrega do <i>software</i> Promob	33
FIGURA 15 – Solicitação	34
FIGURA 16 – Plano de projeto.....	34
FIGURA 17 – <i>Processo</i> de Desenvolvimento do Promob	35
FIGURA 18 – <i>Solicitações</i> de não conformidades	37
FIGURA 19 – Gráfico de <i>resultado</i> operacional.....	39
FIGURA 20 – <i>Gráfico</i> de resultado gerencial.....	39
FIGURA 21 – Gráfico de resultado total.....	39
FIGURA 22 – Requisito de <i>software</i>	50
FIGURA 23 – Modelo de relatório V&V.....	52
FIGURA 24 – Base de conhecimento.....	53
FIGURA 25 – Plano de projeto.....	55
FIGURA 26 – Código fonte.....	55
FIGURA 27 – Documento de requisito implementado	56
FIGURA 28 – Casos de Testes	56
FIGURA 29 – Documento de divulgação	57
FIGURA 30 – Cadastro de Solicitação.....	63
FIGURA 31 – Edição de Solicitação.....	64
FIGURA 32 – Plano de SQA.....	68
FIGURA 33 – Execução do <i>checklist</i> de análise e do <i>checklist</i> de implementação.....	80

FIGURA 34 – Conhecimento para análise	81
FIGURA 35 – Capacidade de criar caso de teste	82
FIGURA 36 – Gráfico alteração de funcionalidade.....	82
FIGURA 37 – Gráfico mudanças na interface.....	83
FIGURA 38 – Gráfico solicitações versus casos de testes	83
FIGURA 39 – Gráfico revisão de casos de testes.....	84
FIGURA 40 – Gráfico retrabalho	85

LISTA DE TABELAS

TABELA 1 - Pesquisa de Satisfação.....	40
TABELA 2 - Plano de SQA	47
TABELA 3 - Plano de V & V	51
TABELA 4 - Métricas do Processo de Desenvolvimento	58
TABELA 5 - Métricas do Processo de SQA.....	58
TABELA 6 - Priorização das características de qualidade.....	59
TABELA 7 - Métricas de usabilidade.....	59
TABELA 8 - Métricas de funcionalidade	60
TABELA 9 - Métricas de eficiência.....	60
TABELA 10 - Plano de Validação.....	62
TABELA 11 - Tabela de documentos a armazenar	66
TABELA 12 - Tabela de riscos	67
TABELA 13 - Controle de Mudanças	68
TABELA 14 - Severidades dos Defeitos da Compilação 4.....	76
TABELA 15 - Tabela da Distribuição dos Erros por etapa da compilação 4	77
TABELA 16 - Tabela da Severidade dos Erros da compilação 4	77
TABELA 17 - Horas de trabalho	78
TABELA 18 - Tabela de requisitos.....	79
TABELA 19 - Tempo de desenvolvimento de compilações.....	79
TABELA 20 - Tempo médio por solicitação	79
TABELA 21 - Requisitos verificados.....	80
TABELA 22 - Experiência do desenvolvedor.....	81
TABELA 23 - Solicitações de Testes.....	84
TABELA 24 - Número de Publicações.....	86
TABELA 25 - Número de defeitos.....	86
TABELA 26 - Impacto dos defeitos	87

LISTA DE ABREVIATURAS

CMMI - (*Capability Maturity Model Integration*) Modelo de Integração de Capacidade e Maturidade

CMMI - DEV - (*Capability Maturity Model Integration for Development*) Modelo de Integração de Capacidade e Maturidade para Desenvolvimento

MPS.BR - Melhoria de Processo de *Software* Brasileiro

ISO - (International Organization for Standardization) Organização Internacional para Normalização

IEC - (Electrotechnical Commission) Comissão Internacional de Eletrotécnica International

IEEE - (*Institute of Electrical and Electronic Engineers*) Instituto de Eletrônica e Engenharia Elétrica

GQM - (*Method Goal-Question-Metrics*) Método Objetivos-Questões-Métricas

PSM - (Method Pratical *Software* Measurement) Método Prático para Medição de *Software*

NBR - Norma Brasileira

SVN - *Subversion*

SQA - Garantia da Qualidade *Software*

SPQA - Plano da Garantia da Qualidade *Software*

ERP - (Enterprise Resource Planning) Sistemas Integrados de Gestão Empresarial

SUMÁRIO

1. INTRODUÇÃO	1
1.1 PROBLEMA DE PESQUISA.....	2
1.2 QUESTÃO DE PESQUISA	3
1.3 OBJETIVOS	3
1.3.1 Objetivos específicos.....	3
1.4 METODOLOGIA.....	3
1.5 ESTRUTURA DO TRABALHO.....	4
2. QUALIDADE DE <i>SOFTWARE</i>	5
2.1 QUALIDADE DE PRODUTO DE <i>SOFTWARE</i>	5
2.2 QUALIDADE DE PROCESSO DE <i>SOFTWARE</i>	8
2.3 Garantia da Qualidade de <i>software</i>	18
2.3.1 Planejamento da qualidade	19
2.3.2 Garantia da qualidade	20
2.3.2.1 Técnicas.....	21
2.3.2.2 Métricas.....	23
2.4.3 Controle da qualidade de <i>software</i>	26
2.4.4 Plano de garantia da qualidade de <i>software</i>	26
2.5 CONSIDERAÇÕES SOBRE SQA.....	29
3. ESTUDO DE CASO PROMOB	31
3.1 ESTRUTURA DA EMPRESA	31
3.2 O PRODUTO PROMOB E SEU PROCESSO DE DESENVOLVIMENTO	33
3.3 QUALIDADE PARA A EQUIPE PROMOB	38
3.4 QUALIDADE PARA O CLIENTE PROMOB.....	40
3.5 CONSIDERAÇÕES	41
4. PROPOSTA DO PLANO DE GARANTIA DA QUALIDADE PROMOB.....	43
4.1 PROPÓSITOS	43
4.2 DOCUMENTOS DE REFERÊNCIA	44

4.3 GESTÃO.....	45
4.3.1 Organização.....	45
4.4 DOCUMENTAÇÃO.....	49
4.4.1 Documentos para V&V, uso e manutenção do <i>software</i>	49
4.4.1.1 Descrição de requisitos de software.....	49
4.4.1.2 Descrição do projeto <i>software</i>	50
4.4.1.3 Plano de V & V (verificação e validação).....	50
4.4.1.4 Relatório de resultados de V&V.....	51
4.4.1.5 Plano de gerenciamento de configuração do <i>software</i>	54
4.4.2 Produtos a serem verificados.....	55
4.5 NORMAS, PRÁTICAS, CONVENÇÕES E MÉTRICAS.....	57
4.6 REVISÕES.....	60
4.7 TESTES.....	61
4.8 RELATÓRIO DE NÃO CONFORMIDADES E AÇÕES CORRETIVAS.....	62
4.9 FERRAMENTAS DE APOIO.....	64
4.10 CONTROLE DE MÍDIA.....	65
4.11 CONTROLE DE FORNECEDORES E CONTRATADOS.....	65
4.12 REGISTROS, MANUTENÇÃO E RETENÇÃO.....	65
4.13 TREINAMENTO.....	66
4.14 GESTÃO DE RISCOS.....	66
4.15 GLOSSÁRIO.....	67
4.16 PROCEDIMENTO DE CONTROLE MUDANÇA DO SQAP E HISTÓRICO.....	67
4.17 CONSIDERAÇÕES DA PROPOSTA.....	68
5. APLICAÇÃO DO PLANO DE QUALIDADE.....	70
5.1 O SQAP NO PROCESSO DE COMPILAÇÃO.....	70
5.2 MUDANÇA NO SQAP.....	73
5.3 APLICAÇÃO DAS MÉTRICAS.....	76
5.4 ANÁLISE DE RESULTADOS.....	78

5.3 CONSIDERAÇÕES DA APLICAÇÃO	87
6. CONCLUSÃO	89
REFERÊNCIAS.....	92
ANEXOS	94

1. INTRODUÇÃO

A história da Garantia da Qualidade de software (SQA) foi desenhada em conjunto com a história da qualidade na fabricação de hardware. Durante a década de 70, pela primeira vez, normas de qualidade foram introduzidas em contratos militares e aplicadas no desenvolvimento de software. Rapidamente se tornaram referência, sendo melhoradas e largamente utilizadas (Pressman, 2006).

Qualidade pode ser definida de uma forma genérica como: "grau no qual um conjunto de características inerentes satisfaz aos requisitos" (NBR ISO 9000, 2005). Software é o conjunto completo ou apenas uma parte dos programas, procedimentos, regras e documentação associada de um sistema computacional de processamento de informação (ISO IEC 2382-1, 1993). A qualidade de software pode ser então definida como "a totalidade de características de um produto de software que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas". (ISO 9126).

Atualmente, órgãos normalizadores, como a ISO - International Organization for Standardization, definem padrões que estabelecem consensos na forma de avaliação de processo e produtos de software. Neste contexto, o modelo de processo de avaliação para o produto de software é descrito pela ISO/IEC 14598. As características de qualidade, o modelo de qualidade e métricas para medição de produto são definições estabelecidas na norma ISO/IEC 9126.

A qualidade do produto reflete a qualidade do processo de desenvolvimento do software (Rocha, 2001). A ISO/IEC 12207, ISO/IEC 15504 e os modelos CMMI e MPS.BR definem, dentre as áreas de processo, uma específica para o processo de Garantia da Qualidade.

A Garantia da Qualidade de software divide-se em três pilares: o planejamento, a garantia e o controle da qualidade. Resultantes deste processo são definidas as atividades, as técnicas, as métricas e todas as definições relacionadas a qualidade de *software*, na dimensão produto e também processo (Bartié, 2002).

A etapa de planejamento gera o Plano de Garantia da Qualidade de Software. Um documento que fornece o caminho para obter a qualidade desejada e guia a equipe para desenvolver produtos de qualidade, em todas as etapas do processo.

Estabelecer o Plano de Garantia da Qualidade para o software Promob, solução para ambientação virtual que lidera o mercado moveleiro nacional e possui licenças distribuídas em diversos países, é necessário para que o produto, que hoje empresta o nome para sua desenvolvedora, atenda a expectativa de qualidade da empresa e de seus clientes.

1.1 PROBLEMA DE PESQUISA

O crescimento e a expansão geográfica de uma empresa geram novas preocupações. Entre estas preocupações está a qualidade dos seus produtos e por consequência com os processos que permitem obter os produtos. Neste contexto, há aproximadamente três anos, a empresa Promob adicionou a etapa de testes, com uma equipe independente, no ciclo de desenvolvimento do Promob.

O Promob é um projeto robusto, com mais de um milhão de linhas de código, desenvolvido e mantido por diferentes equipes. As equipes recebem o planejamento de seu gerente, analisam e constroem o produto requisitado. Após a construção, a equipe de testes e documentação, que é única para todas as unidades de desenvolvimento, de diferentes produtos, é envolvida.

Na fase de testes, o software produzido é validado através de testes manuais, cuja estratégia aplicada é a de caixa preta. As técnicas de caixa preta permitem verificar como os processamentos ocorrem, mas não se o algoritmo inserido no software produz o resultado esperado (BARTIÉ, 2002).

O resultado dos testes são relevantes e importantes, porém limitados já que, aplicados ao final do ciclo, se tornam reativos. Esta é uma fase de gargalo, que provoca uma sobrecarga para as equipes, que faz com que testes sejam negligenciados e correções sejam realizadas em um curto prazo de tempo ou até mesmo não realizadas.

No processo atual um erro no levantamento das necessidades, se torna um erro de análise, que é codificado, e que talvez seja identificado em testes e corrigido, senão somente será identificado em produção. Fatos que culminam para que o produto não seja entregue ao cliente final com a qualidade desejada pela empresa.

1.2 QUESTÃO DE PESQUISA

Como eliminar o gargalo e reduzir os esforços indevidos no ciclo de desenvolvimento do Promob, identificando as falhas em todo o ciclo e não apenas durante a etapa de testes para melhorar a qualidade do *software* Promob?

1.3 OBJETIVOS

Elaborar a proposta de um plano de garantia da qualidade para o *software* Promob. A proposta identificará o conjunto de atividades e artefatos que irão satisfazer os padrões de qualidade, bem como os esforços necessários na prevenção de defeitos, durante as etapas do ciclo de desenvolvimento do produto, sem interferir neste.

1.3.1 Objetivos específicos

Para atingir o objetivo geral apresentado, o trabalho será orientado pelos seguintes objetivos específicos:

- Estudo teórico de bibliografias, cujo tema é a garantia da qualidade de *software*. E também de modelos e normas técnicas relacionadas à qualidade e *software*;
- Diagnóstico do atual ciclo de desenvolvimento do *software*, bem como das características deste produto;
- Proposta de um plano de garantia da qualidade do *software* Promob;
- Aplicação do plano proposto em uma situação real, possibilitando comparar um projeto realizado sem o plano de garantia da qualidade versus um que segue o plano de garantia da qualidade.

1.4 METODOLOGIA

A metodologia deste trabalho tem uma abordagem qualitativa, através de pesquisa bibliográfica. A partir de um estudo de caso, uma proposta de solução será apresentada. A aplicação da proposta em um projeto real permitirá a medição e avaliação dos resultados obtidos.

1.5 ESTRUTURA DO TRABALHO

No capítulo 2 deste trabalho as normas ISO de qualidade de produto de software e qualidade de processo de software são apresentadas, bem como áreas de processo do CMMI e MPS.BR relacionadas ao processo de garantia da qualidade. Os resultados esperados e os atributos destes processos se tornam a referência para a elaboração de um Plano de Garantia da Qualidade. Para tal, técnicas, métricas e todas as atividades necessárias para criação, planejamento e execução do plano são estudadas.

O capítulo 3 descreve o estudo de caso e apresenta a estrutura da empresa Promob, seus processos e o produto Promob. Bem como, as características de qualidade desejadas para o Promob que foram obtidas através da aplicação de uma pesquisa quantitativa.

A elaboração da proposta de um Plano de Garantia da Qualidade, direcionada ao software Promob, é abordada no capítulo 4. No capítulo 5 a aplicação do plano, proposto no capítulo anterior, em um projeto real do *software* Promob, é descrita e os resultados obtidos apresentados.

2. QUALIDADE DE SOFTWARE

O glossário de terminologias de Engenharia de *software* da IEEE - Institute of Electrical and Electronics Engineers – define qualidade de software como sendo o grau de atingimento do software a uma combinação de atributos, sejam eles requisitos especificados ou necessidades e expectativas do cliente.

Para Bartié (2002), qualidade de software é a garantia da conformidade de processos e produtos, ao longo de todas as etapas do desenvolvimento, prevenindo-se contra defeitos. Pressman (2006) define como: “a conformidade com requisitos funcionais e de desempenho explicitamente declarados, normas de desenvolvimento explicitamente documentadas e características implícitas, que são esperadas em todo software desenvolvido profissionalmente”.

É preciso considerar também a subjetividade do termo qualidade, visto que pessoas têm expectativas, grau de satisfação e critérios de avaliação diferenciados para qualidade. Comportamento que torna imprescindível a compreensão, a definição e o consenso da qualidade, em diferentes níveis de abstração (Pressman, 2006).

As próximas subseções tratam das duas dimensões de qualidade de software, qualidade de produto e qualidade de processo, e das normas ISO que se aplicam ao contexto. Os processos de garantia da qualidade do modelo CMMI e MPS.BR e os pilares da qualidade: planejamento, garantia e controle da qualidade. Por fim, as definições para elaborar um Plano de Garantia da Qualidade são apresentadas.

2.1 QUALIDADE DE PRODUTO DE SOFTWARE

Para Rocha (2001), qualidade de *software* pode ser um conjunto de características que permitem avaliar um produto e determinar o grau de qualidade do mesmo. Segundo a NBR ISO/IEC 9126 Qualidade de produto de software, uma característica pode ser detalhada em vários níveis de subcaracterísticas, que formam o conjunto de atributos de qualidade para um produto de software. Esta norma é uma referência básica e suas definições e diretrizes podem ser utilizadas desde a especificação dos requisitos até a avaliação do software.

As características determinadas na norma atendem aos aspectos mais relevantes de produto de software. De forma que cada característica é mais ou

menos relevante, dependendo do propósito para o qual o *software* foi desenvolvido e do ponto de vista considerado.

Ao definir qualidade de software, a NBR ISO/IEC 9126 cita “satisfazer necessidades explícitas e implícitas”. As necessidades explícitas são as condições e objetivos propostos por aqueles que produzem o software. As necessidades implícitas são necessidades subjetivas dos usuários, são também chamadas de qualidade em uso. O modelo definido nesta norma é composto de duas partes, e em ambas a visão do usuário do produto de software é enfatizada.

A primeira parte é definida pela qualidade externa e interna, onde qualidade externa é aquela observada quando o software é executado, tipicamente em testes. Os requisitos de qualidade externa são usados como metas para validação em vários estágios de desenvolvimento e demonstra os efeitos decorrentes da qualidade interna. Os requisitos de qualidade interna podem também ser usados como metas para validação, em vários estágios de desenvolvimento, e para definir estratégias e critérios de avaliação para a verificação, durante o desenvolvimento. Na figura 1, são categorizados os atributos de qualidade de *software* em seis características: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade, as quais são subdivididas em subcaracterísticas.

FIGURA 1 – Modelo de qualidade externa e interna



Fonte: NBR ISO/IEC 9126

A segunda parte da norma, definida como qualidade em uso, é a visão do usuário em relação à qualidade com o *software*. A figura 2 apresenta as 4 (quatro) características deste nível: efetividade, produtividade, segurança e satisfação.

Medidas através dos resultados obtidos no uso do *software* e não mais pelas propriedades do *software*. As qualidades estabelecem uma relação direta de dependências, ou seja, as necessidades de qualidade do usuário incluem requisitos de qualidade em uso em contextos de uso específicos, estas podem ser usadas na especificação da qualidade interna e externa, aplicando características e subcaracterísticas de qualidade do produto de *software*.

FIGURA 2 – Modelo de qualidade em uso



Fonte: NBR ISO/IEC 9126

A série NBR ISO/IEC 9126 possui 4 (quatro) documentos, o 9126-1 que descreve o modelo de qualidade definindo 6 (seis) características e respectivas subcaracterísticas, e contém as diretrizes que descrevem o uso das características de qualidade de software. As demais partes são direcionadas à definição e uso de medições. Em representação hierárquica a norma define as características e subcaracterísticas, as quais podem ser usadas como uma lista de verificação de tópicos relacionados com a qualidade.

A NBR ISO/IEC 9126 define o que deve ser medido. Porém, não é prático atribuir valores diretamente as características de software, papel desempenhado pela ISO/IEC 14598, que apresenta um detalhamento de cada subcaracterística da NBR ISO/IEC 9126, fornecendo métodos para medida, coleta e avaliação da qualidade de produtos de software.

No documento ISO/IEC 14598-3 é abordada a verificação da qualidade de um produto de software, a definição, o acompanhamento e o controle da qualidade durante o desenvolvimento, podendo ser usada na aceitação ou seleção de um produto de *software* ou para a definição e acompanhamento de um processo de avaliação.

A NBR ISO/IEC 9126 e a ISO/IEC 14598 em conjunto tratam das características da qualidade do produto e da avaliação do mesmo, ambas se limitam a avaliar o produto de *software*. Quanto aos processos, que permitem uma organização produzir produtos de qualidade, as normas ISO/IEC 12207 e ISO/IEC 15504 que apresentam as definições e procedimentos.

2.2 QUALIDADE DE PROCESSO DE *SOFTWARE*

O produto final de *software* é o somatório das ações e decisões tomadas durante todo o ciclo de desenvolvimento do mesmo e do investimento na qualidade do processo, que deve contemplar todas as etapas de desenvolvimento a fim de garantir a conformidade de processos e produtos, prevenindo e eliminando defeitos (Bartié, 2002).

A norma internacional ISO/IEC 12207 – Processos de Ciclo de Vida de *software* (2008) - fornece um conjunto de processos para aquisição e fornecimento de produtos e serviços de *software*. Estes processos podem também ser empregados para gerir, desenvolver, usar e melhorar o *software* através do seu ciclo de vida, para isso a ISO estabelece um quadro comum para processos de ciclo de vida de *software*. Seu objetivo é auxiliar no desenvolvimento de *software* e na definição de seus papéis, para que as organizações tenham um melhor entendimento das atividades a serem executadas. Com uma arquitetura de alto nível a norma abrange desde a concepção até a descontinuidade do produto, composta por processos, atividades e tarefas podem ser utilizados para definir, controlar e melhorar os processos do ciclo de vida de *software*.

A estrutura da ISO/IEC 12207 organiza seus processos em três grupos, conforme características e a natureza de cada processo:

- a) Processos fundamentais: envolvem basicamente todas as atividades que a organização executa durante o desenvolvimento, manutenção ou operação de *software*.
- b) Processos de apoio: auxiliam os demais processos contribuindo com a qualidade do projeto de *software*. São processos de apoio porque são executados, quando necessário, por outros processos.

- c) Processos organizacionais: estabelecem e implementam a estrutura constituída pelos processos de ciclo de vida e pessoas envolvidas no desenvolvimento do *software*.

Alinhada com a norma 12207, em 2002 foi publicada a ISO/IEC 15504, cuja origem se dá no projeto SPICE (1993) - *software Process Improvement and Capability Evaluation*, que tinha o objetivo de estabelecer um consenso entre os diversos métodos de avaliação do processo de *software*. Em 1998 foi publicado o relatório técnico da ISO/IEC 15504 definindo uma estrutura para avaliação e melhoria dos processos de engenharia de *software*.

O uso da ISO/IEC 15504 prevê dois contextos: a melhoria de processos e a determinação de capacidade de processo. No primeiro caso, a organização avalia com o objetivo de gerar o perfil dos seus processos. Para identificar os pontos fortes, os pontos fracos e os riscos inerentes aos processos que resultam em um plano de melhorias. Quando a necessidade é avaliar um fornecedor em potencial, a avaliação tem como objetivo determinar a capacidade de processo, pois o perfil de capacidade que permitirá ao contratante estimar o risco associado à contratação do fornecedor.

O documento ISO/IEC 15504-5 desta ISO/IEC - An exemplar software life cycle process assessment model (Um exemplo do modelo de avaliação do processo de ciclo de vida de um software) - disponibiliza um exemplo de modelo de avaliação de processo de software que representa um conjunto de melhores práticas para a engenharia de software. Motivada pelas necessidades de adequação à norma ISO/IEC 15504 e pela evolução da engenharia de software, no ano de 2002 e 2004 a ISO/IEC 12207 recebeu as emendas 1 e 2, respectivamente. Dentre outras atualizações e modificações, a emenda 1 adicionava o processo de SQA. Na publicação atual, ambas as emendas foram incorporadas a norma e o quadro de processos é apresentado conforme figura 3.

Os processos organizacionais são agrupados por práticas de natureza genérica, ou seja, que podem ser usadas por gerentes de projeto ou de processos de um ciclo de vida de *software*. Na categoria de processos de Apoio, onde as maiorias dos processos estão voltados para as atividades da gestão da qualidade. São os processos de apoio voltados a qualidade:

FIGURA 3 – Processos ISO/IEC 12207

Processos 12207		
Processos Fundamentais		Processos de Apoio
Aquisição		Documentação
Fornecimento		Gerência de Configuração
Desenvolvimento	Operação	Garantia da Qualidade
		Verificação
		Validação
		Revisão Conjunta
		Auditoria
	Manutenção	Usabilidade
		Gerência de Resolução de Problemas
		Gerência de Solicitação de Mudanças
		Avaliação do Produto
		Processos Organizacionais
Gerência	Engenharia de Domínio	Melhoria
Gestão de Ativos	Infra-estrutura	
Gestão de Programa de Reúso	Recursos Humanos	

Fonte: ISO/IEC 12207

- a) **Garantia da Qualidade:** que fornece a garantia de que os produtos de trabalho e processos estão em conformidade com os planos e condições pré-definidos.
- b) **Validação:** que confirma que são atendidos os requisitos de um uso específico pretendido para o produto de trabalho de *software*.
- c) **Revisão Conjunta:** que mantém um entendimento comum com os envolvidos a respeito do progresso obtido em relação aos objetivos acordados e ao que deveria ser feito.
- d) **Auditoria:** que determina, de forma independente, a conformidade dos produtos e processos selecionados com os requisitos, planos e contratos, quando apropriado.
- e) **Resolução de Problema:** que assegura que todos os problemas identificados são analisados e resolvidos e que as tendências são identificadas.
- f) **Usabilidade:** que garante que sejam considerados os interesses e necessidades dos envolvidos de forma a proporcionar otimização do suporte e do treinamento, aumento da produtividade e da qualidade do trabalho, melhoria das condições para o trabalho humano e redução das chances de rejeição do sistema por parte do usuário.

- g) Avaliação de Produto: que garante, através de exame e medição sistemáticos, que o produto atende às necessidades especificadas e implícitas dos seus usuários.

Com os mesmos objetivos da ISO/IEC 12207 a qualidade do processo de software, porém com arquitetura em estágios, os modelos CMMI e MPS.BR são guias que estabelecem áreas de processos e almejam produzir software com qualidade.

O CMMI é uma evolução do CMM - Modelo de Maturidade de Capacitação. O CMM foi definido com base nas melhores práticas das indústrias em engenharia e aquisição de *software*, gerenciamento da força de trabalho e engenharia simultânea, e o CMMI define um *framework* para melhoria de processos de *software*, resultante das bem-sucedidas práticas do CMM. A definição dos processos segue os requisitos para um modelo de referência de processo apresentados na ISO/IEC 15504.

A arquitetura do CMMI é composta pela definição de um conjunto de áreas de processo, organizadas em duas representações diferentes: um modelo por estágio, idem ao CMM, e outra como um modelo contínuo, idem à ISO/IEC 15504. Em ambos, a implantação de processos de qualidade segue um amadurecimento gradativo e em patamares. São níveis definidos por um conjunto de características a satisfazer, que determinam qual a capacitação do processo e qual a maturidade que a organização possui para desenvolver *software*. Sendo que, cada nível é um pré-requisito para o seguinte. A diferença do modelo estagiado é que ele estabelece uma sequência pré-determinada para melhoria baseada em estágios que não deve ser desconsiderada. E na representação contínua há flexibilidade na escolha dos processos e no nível de aprofundamento desejado, podendo trabalhar a melhoria de áreas predefinidas.

Cada nível de maturidade consiste em um conjunto de áreas de processos ou PA. Uma PA estabelece metas, específicas e genéricas, a serem satisfeitas por uma sequência de práticas e subpráticas, específicas e genéricas, visando um propósito e um produtos de trabalho. No nível 2, o primeiro degrau a ser conquistado pela organização que adota o CMMI, possui as PAs apresentadas na figura 4 que destaca o processo abordado neste estudo. Este nível consiste no gerenciamento de projeto e traz a expectativa de cumprimento das metas de esforço, prazo e custo. Inicia-se com práticas básicas de gerenciamento, focado em processos rotineiros, onde Garantia da Qualidade de Processo e Produto ou PPQA é uma das áreas.

FIGURA 4 – CMMI nível 2



Fonte: próprio autor

PPQA inclui uma estratégia e procedimentos para avaliar objetivamente processos e produtos, identificando responsáveis, reportando os problemas relacionados e produzindo relatórios que evidenciam as revisões de qualidade e seus resultados. Este processo apoia todas as áreas de processos e deve estar presente nas fases preliminares do projeto para estabelecer planos, processos, padrões e procedimentos que irão adicionar valor, satisfazer seus requerimentos e políticas organizacionais.

A primeira participação da PPQA deve ocorrer no início do ciclo de vida do projeto ao estabelecer o plano de garantia da qualidade, determinando quais processos ou procedimentos são requisitos de valor e, portanto deverão se verificados, e definindo os critérios que serão utilizados na avaliação dos processos que são estabelecidos de acordo com as necessidades de negócio e baseados em padrões aplicáveis, procedimentos e descrição de processos.

São atividades de Garantia da Qualidade de Processo e Produto avaliar os processos, serviços e produtos de trabalho objetivamente, documentar e comunicar os resultados. Ela deve fornecer uma visão objetiva à equipe e à gerência dos processos e produtos de trabalho associados através das seguintes metas e práticas:

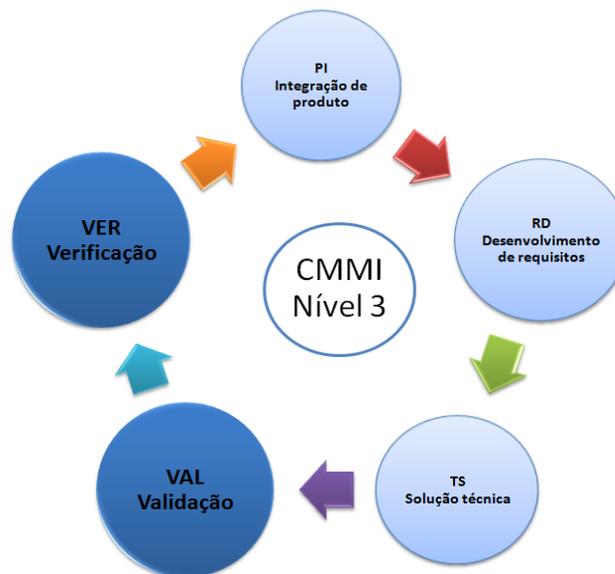
- a) SG 1: Avaliar objetivamente processos e produtos de trabalho
 - SP 1.1: Avaliar processos objetivamente
 - SP 1.2: Avaliar produtos de trabalho e serviços objetivamente

b) SG 2: Prover visão objetiva

- SP 2.1: Comunicar e garantir resolução de problemas discordantes
- SP 2.2: Estabelecer registros

No nível 3 é destacada a importância de estudar, aperfeiçoar e melhorar o processo de desenvolvimento de *software* preocupando-se com a institucionalização deste, que passa a ser visto como um patrimônio da organização. A área de processo de Engenharia possui as áreas de: Desenvolvimento de Requisitos, Solução Técnica, Integração de Produto e, relevantes para este estudo, Verificação e Validação. A figura 5 representa estas áreas e destaca as PAs relevantes para este trabalho.

FIGURA 5 – CMMI nível 3



Fonte: próprio autor

Na Verificação o propósito é o de garantir que o trabalho selecionado atende os requisitos especificados. As metas e práticas específicas são:

a) SG 1: Preparar para verificação

- SP 1.1: Selecionar produtos de trabalho para verificação
- SP 1.2: Estabelecer ambiente para verificação
- SP 1.3: Estabelecer critérios e procedimentos para verificação

b) SG 2: Realizar revisões entre a equipe

- SP 2.1: Preparar para revisão entre a equipe
- SP 2.2: Conduzir revisões entre a equipe
- SP 2.3: Analisar dados de revisão

c) SG 3: Verificar os produtos de trabalho selecionados

- SP 3.1: Realizar verificação
- SP 3.2: Analisar resultados da verificação e identificar ações corretivas.

A área de Validação tem o objetivo de demonstrar que um produto ou componente de produto corresponde ao seu uso esperado quando inserido no ambiente de uso. São metas e práticas específicas desta PA:

a) SG 1: Preparar para validação

- a. SP 1.1: Selecionar produtos para validação
- b. SP 1.2: Estabelecer ambiente para validação
- c. SP 1.3: Estabelecer critérios e procedimentos para validação

b) SG 2: Validar produto ou componente de produto

- a. SP 2.1: Realizar validação
- b. SP 2.2: Analisar resultados da validação

O CMM teve origem a partir de um pedido do Departamento de Defesa dos Estados Unidos para o SEI - *software Engineering Institute*. A SOFTEX - Associações para Pro São resultados esperados deste processo e relevantes para este trabalho:

- a) 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;
- b) GQA 2: A aderência dos processos executados às descrições de processo, padrões e procedimentos é avaliada objetivamente;
- c) GQA 3: Os problemas e as não-conformidades são identificados, registrados e comunicados;
- d) 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.

O CMM teve origem a partir de um pedido do Departamento de Defesa dos Estados Unidos para o SEI - *Software Engineering Institute*. A SOFTEX - Associações para Promoção da Excelência do *Software* Brasileiro - é uma iniciativa nacional, que coordena ações, cujo objetivo é aumentar a competitividade da indústria brasileira de *software*, nos mercados interno e externo, através de programas de qualificação de profissionais e de melhoria e avaliação de processos e

produtos de *software* brasileiros. Produto deste esforço, o programa MPS.BR acrônimo de Melhoria de Processos do *Software* Brasileiro, foi criado em 2003 e tem sua base no CMMI-DEV e nas normas ISO/IEC 12207 e ISO/IEC 15504.

Cada nível de maturidade do MPS.BR possui suas áreas de processo, onde são analisados os processos fundamentais, organizacionais e os processos de apoio. A definição dos processos do MPS.BR declara o propósito e os resultados esperados de sua execução, que podem ser um produto de trabalho ou uma mudança significativa no processo. À medida que a organização evolui nos níveis de maturidade, que inicia em G e progride até o nível A, estabelece patamares de evolução que e caracterizam estágios de melhoria da implementação de processos na organização.

No nível F, conforme figura 6, o processo de Garantia da Qualidade com o propósito de assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os planos, procedimentos e padrões estabelecidos devem ser implementado.

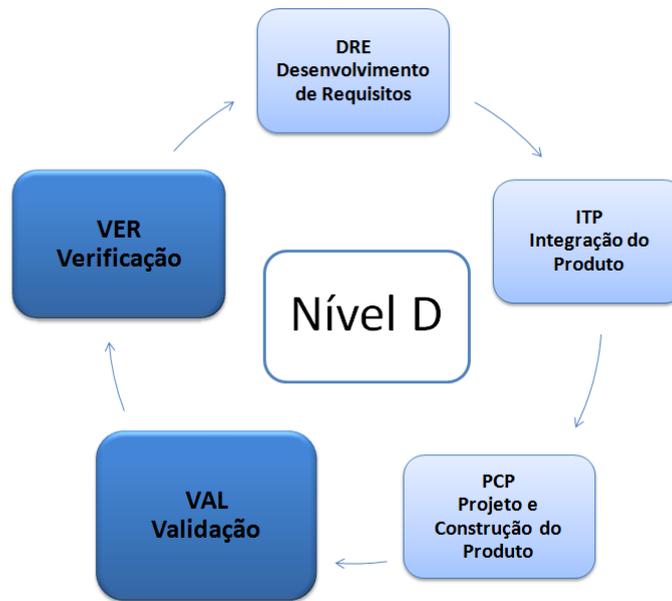
FIGURA 6 – MPS.BR nível F



Fonte: próprio autor

No nível D, o MPS.BR apresenta o processo VAL - Validação e VER - Verificação, conforme figura 7. Validação tem como propósito confirmar que um produto ou componente do produto atenderá a seu uso pretendido quando colocado no ambiente para o qual foi desenvolvido.

FIGURA 7 – MPS.BR nível D



Fonte: próprio autor

Para isso define como resultados esperados:

- a) VAL1: Produtos de trabalho a serem validados são identificados para serem validados ao longo do projeto;
- b) VAL2: Uma estratégia de validação é desenvolvida e implementada, estabelecendo cronograma, participantes envolvidos, métodos para validação e qualquer material a ser utilizado na validação. Tem como objetivo garantir que as atividades de validação sejam planejadas com a definição de procedimentos, infraestrutura necessária, cronograma, recursos e responsabilidades. Os métodos a serem usados nas atividades de validação também devem ser identificados;
- c) VAL3: Critérios e procedimentos para validação dos produtos de trabalho a serem validados são identificados e um ambiente para validação é estabelecido, cujo objetivo é garantir que os critérios e procedimentos a serem utilizados para a validação foram identificados e que foi estabelecido um ambiente para validação, semelhante ao ambiente operacional;
- d) VAL4: Atividades de validação são executadas para garantir que o produto esteja pronto para uso no ambiente operacional pretendido.
- e) VAL5: Problemas são identificados e registrados. Estes problemas devem ser acompanhados até sua conclusão;

- f) VAL6: Resultados de atividades de validação são analisados e disponibilizados para o cliente, ou seu representante na execução das atividades, e outras partes interessadas, e;
- g) VAL7: Evidências de que os produtos de *software* desenvolvidos estão prontos para o uso pretendido são fornecidas. O processo de VER busca confirmar que cada serviço e/ou produto de trabalho do processo ou do projeto atende apropriadamente os requisitos especificados. Garantia a ser obtida através dos seguintes resultados:
 - a) VER1: Produtos de trabalho a serem verificados são identificados. Para atender a este resultado esperado devem-se analisar os produtos de trabalho que serão produzidos ao longo do projeto e selecionar aqueles a serem verificados;
 - b) VER2: Uma estratégia de verificação é desenvolvida e implementada, estabelecendo cronograma, revisões envolvidos, métodos para verificação e qualquer material a ser utilizado na verificação. Este resultado envolve definir uma estratégia de verificação descrevendo os procedimentos, a infraestrutura necessária e as responsabilidades pelas atividades de verificação;
 - c) VER3: Critérios e procedimentos para verificação dos produtos de trabalho a serem verificados são identificados e um ambiente para verificação é estabelecido. O que implica na definição dos critérios e procedimentos que serão utilizados para a verificação de cada produto de trabalho e na preparação do ambiente para verificação, disponibilizando ferramentas, recursos de hardware, infraestrutura de rede e outros recursos necessários à execução das atividades planejadas;
 - d) VER4: Atividades de verificação, incluindo testes e revisões por pares, são executadas. Esta busca garantir que as atividades de verificação são executadas conforme planejado, o que inclui, obrigatoriamente, a realização de revisão por pares e testes;
 - e) VER5: Defeitos são identificados e registrados. Trata-se de documentar para registrar os defeitos identificados durante a execução da verificação, e;
 - f) VER6: Resultados de atividades de verificação são analisados e disponibilizados para as partes interessadas. Para tal, a análise dos resultados obtidos em cada atividade de verificação deve ser realizada disponibilizando os resultados para as partes integrantes do processo.

O processo de Garantia da qualidade, o seu planejamento, definições de estratégias, técnicas e critérios, métricas e ferramentas, resultando no plano de garantia da qualidade para que o controle da qualidade seja executado.

2.3 GARANTIA DA QUALIDADE DE SOFTWARE

Segundo Sommerville (2007), SQA – Garantia da Qualidade de *software* - é o processo de definição de como a qualidade de *software* pode ser atingida e como a organização de desenvolvimento sabe que o *software* possui o nível de qualidade necessário. O processo de garantia de qualidade está, principalmente, relacionado à definição e seleção de padrões que devem ser aplicados ao processo de desenvolvimento de *software* ou ao produto de *software*.

Pressman (2006) define como um processo de avaliação que visa garantir que cada produto de trabalho, bem como o processo ao longo de todo o ciclo de desenvolvimento está em conformidade aos requisitos e padrões estabelecidos. Acrescenta que pode atuar como um facilitador para antecipar a detecção de falhas ou não conformidades ao longo do ciclo de vida do desenvolvimento, minimizando o custo para se corrigir um erro, visto que este aumenta à medida que as fases do projeto são finalizadas.

O guia MPS.BR (2011) - Melhoria de Processos do software Brasileiro - estabelece que a Garantia da Qualidade deve contemplar tanto a gerência quanto a construção dos produtos de trabalho. Portanto, é o principal objetivo da Garantia da Qualidade disseminar uma cultura de que a qualidade não deve ser testada e avaliada somente ao término do desenvolvimento do produto e sim, em todas as fases do desenvolvimento.

Bartié (2002) complementa dizendo que não é possível obter qualidade em uma única etapa do ciclo de vida do software, todas as etapas devem ser contempladas com atividades, que visem a garantia da qualidade de software. Pressman (2006) explica que há uma distinção clara entre um erro, que é um problema encontrado antes da entrega do software ao cliente, e um defeito, que é o problema identificado após a entrega do produto ao cliente. O momento em que o problema foi encontrado é determinante no impacto econômico, psicológico e humano. SQA atua nas duas frentes, através de técnicas de prevenção, como as

revisões, apoiada pelas práticas de verificação e de técnicas de detecção, como testes, apoiada pelas práticas de validações.

O processo de garantia da qualidade de software possui três subprocessos, os quais Bartié (2002) chama de Pilares da Qualidade. São eles: Planejamento da qualidade, Garantia da qualidade e Controle da qualidade. O documento Plano de Garantia da Qualidade é o guia de SQA e o primeiro produto a ser produzido pelo processo. Os pilares e o plano serão abordados nas subseções seguintes.

2.3.1 Planejamento da qualidade

O objetivo do planejamento da qualidade é identificar os atributos de qualidade relevantes para o produto, os padrões organizacionais apropriados e o processo de desenvolvimento a ser adotado para o projeto, e determinar como satisfazê-los. Para isso a política da qualidade, os procedimentos, os padrões e a descrição do produto devem ser utilizados, como ilustra a figura 8.

FIGURA 8 – Etapas do planejamento de qualidade



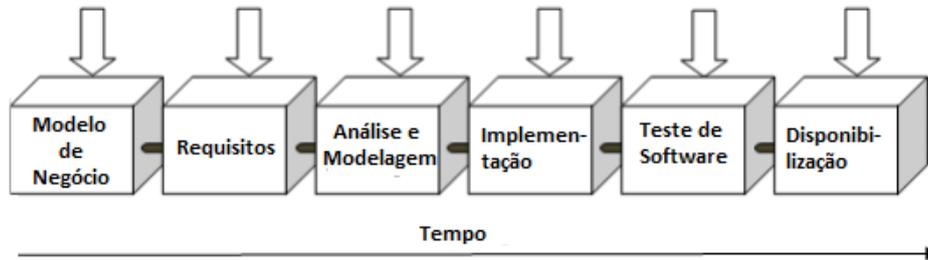
Fonte: próprio autor

A política compreende as intenções e diretrizes de uma organização, formalmente expressadas pela alta gerência. O escopo do projeto gera as entradas para o planejamento da qualidade determinando os produtos a serem entregues e os objetivos do projeto. Os padrões devem ser considerados no planejamento incluindo processos, modelos e técnicas definidas pela organização, além de padrões regulamentares externos, caso existam. Por fim, considera-se o entendimento das características do *software* a ser gerado.

A garantia da qualidade atua sobre todas as atividades que englobam o desenvolvimento do produto. Portanto, seu planejamento deve contemplar todas as etapas do ciclo de desenvolvimento do produto de *software*, como mostra a figura 9. Desta forma, é possível descobrir um maior número de defeitos antecipadamente,

tornando o processo mais estável, menos caótico e reduzindo os custos (Bartié, 2002).

FIGURA 9 – Processo de garantia da qualidade de *software*

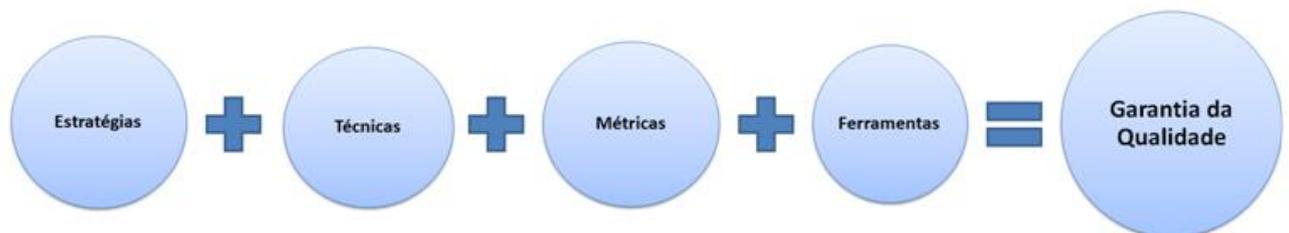


Fonte: Bartié, 2002

2.3.2 Garantia da qualidade

A garantia da qualidade é um conjunto de atividades estruturadas e sistemáticas, planejadas para auditar e relatar a não conformidade dos processos e produtos, ao longo de todo o ciclo de vida do *software*, a fim de satisfazer o modelo de qualidade adotado pela organização. Para isso a estratégia deve ser definida, bem como as técnicas, metodologias e ferramentas de apoio ao desenvolvimento, como mostra a figura 10.

FIGURA 10 – Etapas da garantia da qualidade



Fonte: próprio autor

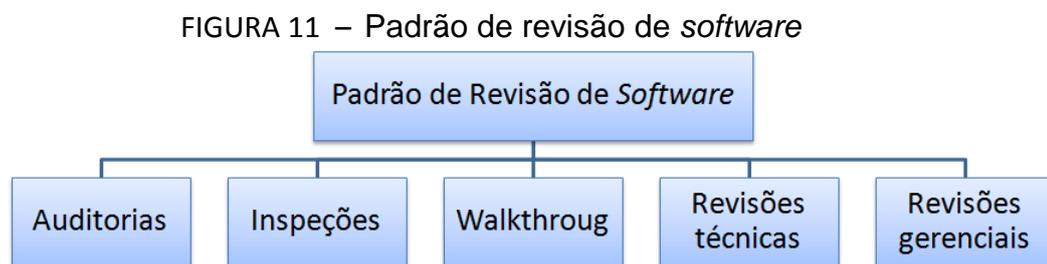
A definição das estratégias é o momento em que a organização define técnicas, a periodicidade de avaliação dos processos organizacionais e do produto e estratégias de correção das não conformidades a serem adotadas para trabalhar e produzir com qualidade. Técnicas e Métricas são abordadas nas próximas seções.

2.3.2.1 Técnicas

Para garantir a qualidade do produto e do processo da organização, durante as fases de desenvolvimento do produto técnicas estáticas, que avaliam a documentação do projeto e os produtos de trabalho independentes da execução do código, e dinâmicas, que avaliam a partir da execução de código, devem se usadas. As mesmas também são chamadas Métodos Estruturados de Verificação e Métodos Estruturados de Validação, respectivamente, por Bartié (2002) que recomenda o uso de revisões para as atividades que focalizam documentações, que é a mesma citada por Pressman (2006), e Auditoria para avaliar as atividades.

A revisão de software e avaliação são as técnicas estáticas mais utilizadas. A avaliação é uma forma de analisar um processo em comparação a modelos, normas ou boas práticas buscando identificar a capacidade e identificar melhorias. O CMMI e a ISO/IEC 15504 são modelos de avaliação. A técnica de revisão de software estabelece procedimentos de avaliação de software e de toda a documentação gerada durante o desenvolvimento do produto, garantindo que todas as atividades estão sendo conduzidas adequadamente e reduzindo eventuais falhas que possam prejudicar o desenvolvimento do produto (Bartié, 2002).

No padrão IEEE 1028 – 1997 Standard for *software Reviews* (Padrão de Revisão de *software*) as revisões são classificadas em 5 tipos: Auditorias, revisões técnicas, inspeções, *Walkthrough* e revisões gerenciais, conforme representado na figura 11.



Fonte: próprio autor

a) Auditorias:

Avaliam o processo e o produto de *software*, através de critérios objetivos, para identificar a conformidade em relação aos padrões e procedimentos preestabelecidos pela organização. São avaliações formais, executadas por pessoas que não tenham relacionamento direto com o que está sendo

auditado. São utilizadas para avaliação, entrevistas e documentos. As não conformidades detectadas devem ser convertidas em ações corretivas, o auditor e a unidade auditada, são responsáveis por esta. Um relatório final, contendo as conformidades, não conformidades e ações corretivas com responsáveis e prazos, finaliza a Auditoria.

b) Inspeções:

Também são avaliações formais, possuem procedimentos e papéis bem definidos. Avaliam produtos de *software* para detectar falhas, anomalias ou desvios significativos em relação aos documentos do projeto, ao produto de *software* e aos padrões preestabelecidos. Uma inspeção é executada por especialistas, através da utilização de *checklists*, que podem ser aplicados em diversas etapas ao longo do ciclo de vida de *software*. O resultado da inspeção é a aceitação do produto, aceitação condicional ou rejeição, sendo necessárias ações corretivas e novas inspeções.

c) *Walkthrough*:

Avaliações menos formais que tem como objetivo educar o seu público-alvo sobre a importância da realização de avaliações nos produtos de *software*. Não tem papéis bem definidos, nem exige treinamento para o avaliador. A forma de aplicar e os resultados não diferem da inspeção. *Walkthrough* é recomendado para organizações que ainda não trabalham a revisão de seus produtos de *software*.

d) Revisões Técnicas:

O objetivo da revisão técnica é verificar a conformidade de um produto contra os padrões estabelecidos, garantir a integridade do produto em relação às mudanças, e avaliar em relação ao objetivo pelo qual foi proposto. É uma revisão formal, portanto com planejamento bem definido e uma equipe de avaliação qualificada. Os artefatos a serem avaliados devem ser definidos e um período de tempo para avaliação por parte da equipe. Os resultados são expostos em reunião e recomendações ou alternativas são sugeridas e planejadas caso haja não conformidades.

e) Revisões Gerenciais:

Usadas para avaliar a eficácia dos processos e padrões aplicados no desenvolvimento do produto de *software*. Podem ser utilizadas também para monitorar o andamento do projeto. O gestor envolvido no processo deve

participar. A execução assemelha-se às revisões técnicas e visa identificar adequações e inadequações nos padrões e procedimentos gerenciais. Os produtos de *software* que podem ser utilizados para uma revisão gerencial são: A lista de não conformidades do projeto, os documentos de Auditorias, o plano do projeto, o plano de contingência e risco, o plano de Garantia da Qualidade, entre outros.

As técnicas dinâmicas são representadas pelos testes. Testes são executados com o objetivo de identificar e encontrar erros no produto de *software* gerado e representam a última oportunidade de detectar erros antes da entrega. Entretanto, devem-se definir quais estratégias e a priorização das categorias de testes para obter eficácia na detecção de falhas do produto.

As principais estratégias de testes de *software*, segundo Pressman (2006), envolvem:

- a) Testes de unidade: visa testar individualmente a menor unidade de projeto de *software*;
- b) Testes de integração: visa testar o relacionamento entre as diversas unidades integradas, pois não há garantias de que unidades testadas em separado funcionarão em conjunto;
- c) Testes de validação: visa garantir que o *software* funciona de acordo com os critérios de validação comumente especificados na definição dos requisitos de *software*, ou que funciona de acordo com as expectativas e necessidades dos clientes. Envolve testes alfa e beta. Ambos são realizados por clientes, no alfa monitorado e em ambiente controlado e no beta sem a supervisão e no ambiente de produção;
- d) Testes de sistema: conjunto de testes cujo objetivo é colocar à prova todo o sistema. Esta estratégia envolve a realização de testes de recuperação, segurança, *stress* e desempenho para garantir as funcionalidades e a integração do *software*.

2.4.2.2 Métricas

A definição de métricas para garantia da qualidade ocorre ao definir padrões de medidas e coleta para o produto e para o processo com o objetivo de entender, controlar e gerenciar todo o processo de desenvolvimento de *software*. O que resulta

na identificação de efetividade do processo e melhorias, quando necessário (MPS.BR, 2011).

Sommerville (2007) define métrica de *software* como um tipo de medida atribuído a um sistema, documento ou até mesmo ao processo de desenvolvimento de *software*, ou seja, a partir da conversão de um atributo em valor numérico ou símbolo, este é comparado aos padrões de qualidade ou regras definidas, visando prover e identificar melhorias ou identificar não conformidades.

Uma medida é conceituada por Pressman (2006) como a “indicação quantitativa da extensão, quantidade, dimensão, capacidade ou tamanho de algum atributo de um produto ou processo”. Embora utilizados sob o mesmo contexto, medida, medição, métrica e indicador possuem diferenças. Medição é o ato de determinar uma medida e métrica é o relacionamento de medidas individuais para se avaliar o grau em que um sistema se encontra em relação a um determinado atributo. Um indicador é uma métrica ou uma combinação de métricas que visa fornecer informações para compreensão do processo, produto ou projeto.

Os aspectos a serem medidos para Garantia da Qualidade de *software* devem considerar os processos de desenvolvimento e produtos de *software*. Métricas distintas devem ser aplicadas para a medição de produtos e de processos. Sommerville (2007) divide as métricas em de controle e preditivas. Métricas de controle são voltadas a processos e preditivas ao produto de *software*. As métricas preditivas ou de produto visam obter dados quantitativos a partir das características do próprio *software*, estas o mesmo autor subdivide em duas classes, dinâmicas e estáticas.

Métricas dinâmicas são coletadas com o programa em execução e avaliam a eficiência e a confiabilidade do *software*, por exemplo. Métricas estáticas são coletas por medições realizadas em documentos e permitem avaliar a complexidade e facilidade de compreensão e manutenção do sistema. Por exemplo, na fase de análise devem ser utilizadas métricas que avaliem a qualidade da especificação de requisitos. Já na fase de construção do produto, as medições envolvem o código fonte e na fase de testes do produto avaliam a cobertura (Pressman, 2006; Bartié, 2002).

Mais fáceis de serem medidas as métricas de processo são utilizadas para obter dados quantitativos sobre o processo de *software*. Esta medição fornece dados para a melhoria do processo. Normalmente, este tipo de métrica quantifica

atributos relacionados ao tempo, ou seja, medição para identificar em quanto tempo atividades são concluídas, esforço ou recursos necessários para um processo específico, e número de ocorrências que são incidências de eventos específicos.

Importantes empresas como a Hewlett-Packard são citadas por Sommerville (2007) como exemplo de organizações que adotaram programa de métricas, através de programas de gerenciamento de qualidade de *software*. Estas abordadas através da medição de defeitos e processos de validação e verificação.

Contudo, Sommerville (2007) conclui que o uso de métricas é pouco comum, visto que seus benefícios não são bem definidos, e justifica sua conclusão argumentando que os motivos deste comportamento são: processos de *software* definidos precariamente e *software* não suficientemente aperfeiçoados para aplicarmos métricas, para o segundo motivo o autor afirma que com frequência, não é possível medir os atributos de qualidade de *software* diretamente e explica que se pode utilizar algum atributo interno do *software* e supor que exista uma relação entre o que é possível medir e o que se deseja saber.

Para medir se faz necessário primeiramente definir o que medir, o padrão IEEE 1061 – 1998 *Standard for a software Quality Metrics Methodology* (Padrão de Metodologias para Métricas da Qualidade de *software*) cita dois métodos que podem ser úteis para o planejamento do trabalho de medição. São eles: o Método GQM - *Goal-Question-Metrics* - e o Método PSM (*Practical software Measurement*). Em seguida, componentes a serem avaliados devem ser selecionados, esta seleção pode ser representativa dos componentes que se deseja avaliar ou pela importância destes para o produto.

A precisão de uma avaliação de qualidade depende muito da escolha das métricas. Conforme a norma ISO/IEC 14598 algumas características das métricas podem aumentar a confiabilidade das mesmas. São elas:

- a) Significância: a métrica deve agregar informação sobre o comportamento do *software* ou sobre suas características de qualidade;
- b) Custo e complexidade: a aplicação da métrica deve ser econômica e tecnicamente viável;
- c) Reproducibilidade e Repetibilidade: capacidade de uso da métrica em diferentes momentos e/o produtos deve produzir resultados idênticos;
- d) Validade: demonstrar a corretude e precisão ou a margem de erro dos resultados da medição deve ser possível;

- e) Objetividade e Imparcialidade: os resultados da medição devem ser objetivos e não deve ser tendenciosa.

A norma NBR ISO/IEC 9126-2 fornece métricas para avaliar o produto de *software*. Ela determina desdobramentos das 6 características que apresenta, o primeiro deles é define subcaracterísticas e o seguinte são os atributos. A partir destes pode-se realizar medições e determinar, por exemplo que: “O tempo de uso do produto de *software* até que se tenha domínio operacional do mesmo deverá ser inferior a 20 horas”, facilitando a difícil tarefa de obter a avaliação de produto de *software*. (Koscianski, et. al., 1999).

Muito específicas e preciosistas, muitas das métricas da norma exigem que o *software* seja desenvolvido pensando em medi-las, por exemplo, na característica eficiência, a subcaracterística, utilização de recursos, apresenta a medição de “utilização máxima de transmissão” referenciando o barramento.

Para selecionar as métricas Pressman (2006) orienta que para equipes de até 20 (vinte) pessoas de *software*, é irracional e até inviável esperar programas abrangentes de métricas, o que não exclui a recomendação medir e aplicar métricas para aperfeiçoar processo, qualidade e pontualidade dos produtos.

2.4.3 Controle da qualidade de *software*

O controle de qualidade envolve as revisões e testes, aplicados durante o ciclo de desenvolvimento de um produto, para avaliar a qualidade do mesmo, a fim de que ele satisfaça os níveis de qualidade desejados.

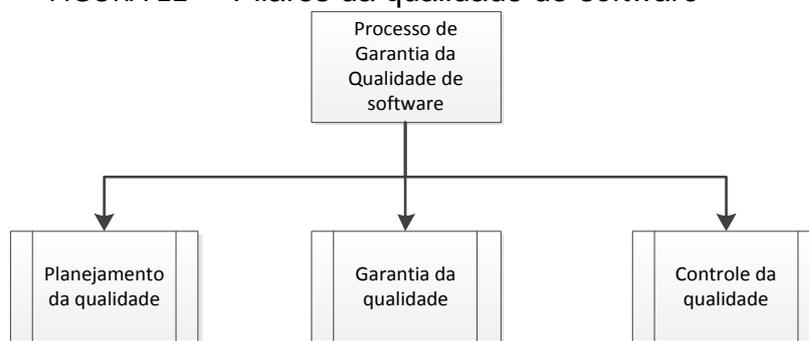
Para Pressman (2006) o conceito-chave do controle de qualidade é que todos os produtos de trabalho têm especificações definidas e mensuráveis, que permite comparar o resultado de cada processo e construir o ciclo de retroalimentação no processo de trabalho, permitindo minimizar os defeitos produzidos.

2.4.4 Plano de garantia da qualidade de *software*

O processo de SQA reúne os objetivos de planejamento, garantia e controle da qualidade, como mostra a figura 12. Para alcançar estes objetivos e as qualidades desejadas para o produto a ser desenvolvido e com aderência ao processo de desenvolvimento, estes são registrados em um documento, o Plano de garantia da Qualidade.

Conforme a ISO/IEC 12207, o plano de garantia de qualidade define os artefatos sob controle de garantia de qualidade e suas formas de verificação. Os processos de verificação, validação, revisão e auditoria estão vinculados ao processo de qualidade e suas atividades geram evidências de qualidade. O encaminhamento de problemas e não conformidade é realizada por meio do processo de resolução de problemas. No MPS.BR diz que entre outros, o plano deve conter os marcos ao longo do ciclo de vida do projeto nos quais atividades de Garantia da Qualidade deverão ser executadas.

FIGURA 12 – Pilares da qualidade de *software*



Fonte: Bartié, 2002.

Sommerville (2007) diz que um plano de qualidade estabelece as qualidades desejadas para o produto a ser desenvolvido e como essas qualidades devem ser avaliadas. Também deve selecionar os padrões organizacionais que sejam apropriados a um determinado produto e processo de desenvolvimento a ser adotado e definir os atributos de qualidade mais significativos para o produto em questão e recomenda que este documento seja descrito de forma sucinta.

Para Bartié (2002) é a visão comum sobre todos os recursos e esforços a serem utilizados para garantir a qualidade durante todo o ciclo de desenvolvimento de *software*, que envolve todos os integrantes do projeto. Pressman (2006) reforça que o Plano de Garantia da Qualidade rege todas as atividades a serem executadas pelo grupo ou área de SQA:

- a) Avaliações, Auditorias e revisões a serem realizadas;
- b) Normas aplicáveis ao projeto;
- c) Procedimento para reporte e acompanhamento de não conformidades;
- d) Plano de comunicação de tais não conformidades;

- e) Documentos a serem produzidos e as responsabilidades do grupo ou área de SQA. Sendo, portanto, um roteiro que contempla o conjunto de atividades da estratégia de uma equipe de qualidade.

O padrão IEEE Std 730 (*Standard for software Quality Assurance Plans*) descreve uma estrutura de tópicos, a qual chama de seções, para o documento Plano de Garantia da Qualidade. Este modelo possui seções que:

- a) Especificam objetivos e escopo do plano;
- b) Atribuem às responsabilidades a equipe;
- c) Determina os produtos e processos a serem avaliados;
- d) Apresenta os documentos de referência, as normas, os padrões e as políticas adotadas pela organização
- e) Apresenta técnicas e métricas para avaliação, as ferramentas de apoio entre outras.

Pressman (2006) faz referência ao modelo da IEEE 730. Já Bartié (2002) sugere uma estrutura onde têm um plano mestre de verificação e outro de validação, subordinados ao plano de qualidade. Estes têm como objetivo estruturar os processos de verificação e validação, respectivamente, estabelecendo a visão da equipe, uniformizando conhecimentos, experiências e expectativas para os que integram o processo de desenvolvimento de *software*. O documento do plano de qualidade, além dos planos mestres, seria composto por outros itens:

- a) Propósito do plano;
- b) Apresentação do processo de verificação e validação, e gerenciamento de SQA;
- c) Apresentação da equipe de revisões e Auditorias, e equipe de testes;
- d) Histórico das experiências passadas, principais documentações a serem empregadas;
- e) Referências a ferramentas, técnicas e metodologias;
- f) Referências a padrões, práticas, convenções e métricas;
- g) Treinamento, entre outros.

Ao final da elaboração, o SQAP – Plano de Garantia da Qualidade de *software* – deve ser revisado e acordado com os responsáveis pelas áreas onde a Garantia da Qualidade atuará. O consenso de todos os envolvidos e o comprometimento com a Garantia de Qualidade é decisivo para alcançar os objetivos para o qual o plano foi proposto.

2.5 CONSIDERAÇÕES SOBRE SQA

O desafio de uma cultura de não tolerância a erros é proposto pela SQA, possibilitando a avaliação e a identificação prematura de defeitos em produtos intermediários do *software*.

O modelo de qualidade definido na NBR ISO/IEC 9126 permite identificar os requisitos de qualidade e direcionar corretamente os esforços de *software* e critérios para garantia de qualidade na dimensão de produto.

A ISO/IEC 14598 – Avaliação de Produto – pode ser utilizada juntamente com a NBR ISO/IEC 9126 fornecendo os requisitos e recomendações para implementação do processo de avaliação de produtos de *software*.

A dimensão qualidade de processo é abordada através da ISO/IEC 12207, em conjunto com a ISO/IEC 15504, e nas práticas do CMMI e MPS.BR. A ISO/IEC 12207 descreve uma estrutura para o ciclo de vida do *software* e nos processos de apoio, os quais devem ser utilizados por qualquer um dos demais processos, afinal a qualidade deve estar presente em todo o ciclo de vida de um produto de *software*, estabelece as definições relacionadas aos requisitos de qualidade.

Apesar das particularidades, normas e modelos, estão alinhadas em definições, propósitos e resultados para o processo de garantia da qualidade, que tem como objetivo executar os processos de forma aderente aos padrões e procedimentos adotados pela organização. Nos processos de Validação e Verificação, que apoiam a avaliação dos produtos intermediários e final e suportam outras atividades que devem estar contidas no plano de garantia da qualidade a fim de gerar produtos de trabalho com qualidade.

A partir destas normas e modelos, pode-se constatar que o processo de Garantia da Qualidade deve:

- a) Desenvolver, documentar, implementar e manter o Plano de Garantia da Qualidade;
- b) Atribuir responsáveis para o planejamento e execução das atividades deste processo;
- c) Documentar e garantir a solução de não conformidades;
- d) Avaliar de forma objetiva processos e produtos de trabalho, durante todo o ciclo de desenvolvimento;
- e) Comunicar resultados obtidos.

Os processos de Validação e Verificação suportam as atividades de:

- a) Seleção de artefatos e produtos a serem verificados e validados;
- b) Seleção dos critérios e procedimentos que serão utilizados;
- c) Definição de métodos de verificação e validação;
- d) Planejamento dos testes;
- e) Execução da verificação e/ou validação;
- f) Definição da equipe responsável.

Os processos de verificação e validação estão vinculados ao processo de qualidade de *software*, gerando evidências de qualidade através da avaliação dos produtos intermediário do processo de desenvolvimento.

O processo de garantia da qualidade precisa ser estruturado para que o objetivo maior deste trabalho, o SPQA, possa ser alcançado. SQA possibilita que defeitos sejam identificados em todas as etapas do processo de desenvolvimento, estabelecendo atividades, técnicas e métricas para prover a garantia da qualidade dos produtos e processos da organização.

As técnicas de revisão, Auditorias e testes geram relatórios de avaliação, não conformidades e ações corretivas para processo e produto, que devem ser devidamente registradas e gerenciadas pela Garantia da qualidade para assegurar a resolução de problemas e contribuir com melhorias para o processo e produto. As métricas têm por objetivo mensurar a qualidade dos produtos de trabalho e do processo ao longo das fases de desenvolvimento. Estas ações devem contribuir para evitar gargalos na etapa de testes e aumentar a qualidade do produto.

3. ESTUDO DE CASO PROMOB

Para propor um plano de qualidade é necessário conhecer a empresa, o processo de desenvolvimento e as características de qualidade desejadas para os produtos. Como estas definições não estavam formalizadas na empresa foi necessário diagnosticar e modelar o processo de desenvolvimento da U.D. – Unidade de Desenvolvimento. Em paralelo, foram levantadas as características de qualidade desejáveis para o produto para o qual o plano de garantia da qualidade será proposto, o Promob.

Nas seções a seguir a estrutura da empresa, o produto Promob e o processo de desenvolvimento do mesmo são apresentados. Bem como, as pesquisas que revelaram as características de qualidade desejadas para o produto.

3.1 ESTRUTURA DA EMPRESA

A Promob é uma empresa de tecnologia, inicialmente especializada no desenvolvimento de *software* para ambientação virtual. Atualmente, dispõe de soluções para concepção e fabricação de móveis, ERP, gestão de lojas e gestão empresarial. Criada em 1994, a empresa projetou-se rapidamente como líder no setor moveleiro, mercado no qual se especializou, tornando-se sinônimo de *software* para *leiout* no mercado brasileiro.

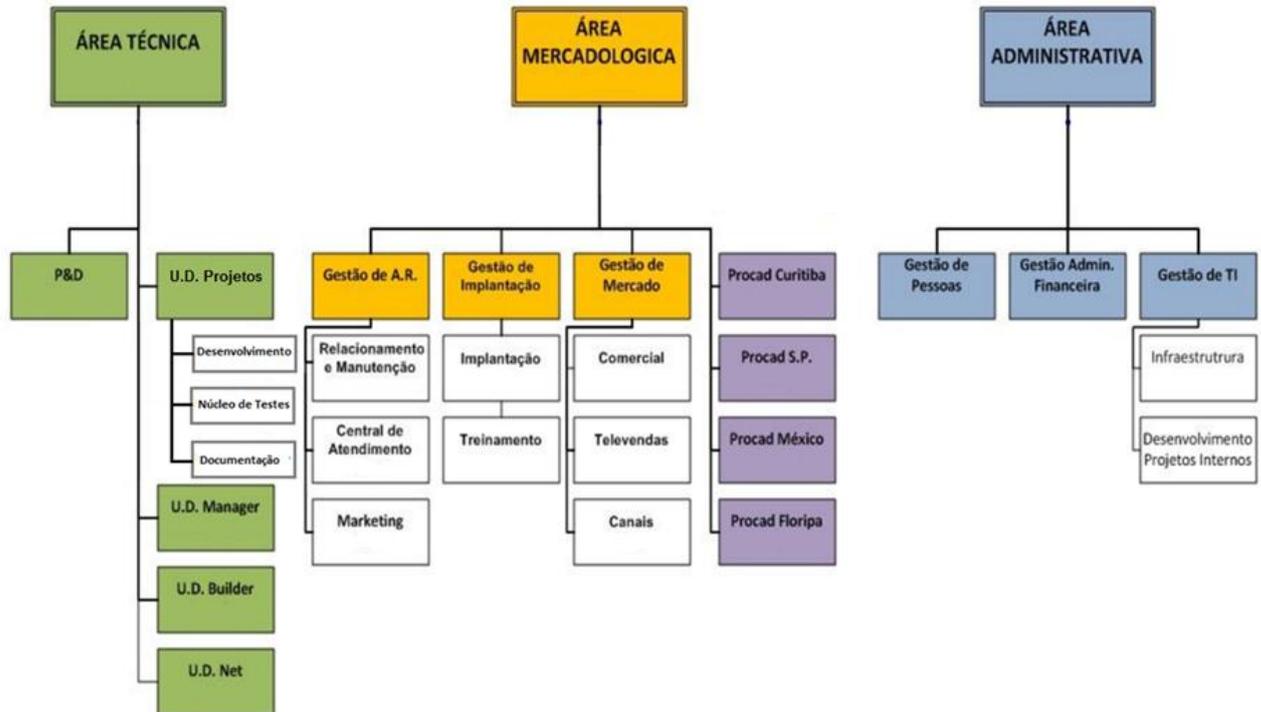
Com sede em Caxias do Sul-RS e filiais em capitais, como: Belo Horizonte, Curitiba, Florianópolis, São Paulo e no exterior, nos Estados Unidos, Argentina e México. A Promob *software* Solutions é referência no país no segmento no qual atua e vem se consolidando também no mercado mundial.

Na matriz, sediada em Caxias do Sul – RS, estão centralizadas as atividades de pesquisa e desenvolvimento de produtos. O organograma da Promob matriz possui 3 áreas: Técnica, Administrativa e Comercial. A área técnica é dividida em U.D.s - Unidade de Desenvolvimento – que são especializadas por tipo de produto: soluções para projetos (U.D. – Projetos), soluções de fabricação (U.D. Builder), soluções web (U.D. Net) e gestão de lojas (U.D. Manager), representado na figura 13.

O Promob foi o primeiro produto desenvolvido pela empresa, e o que permitiu a mesma chegar ao patamar onde se encontra atualmente. Passados 17 anos de

atuação no mercado ao qual atende, o moveleiro, o produto se tornou tão forte que no ano de 2011 a empresa Procad, mudou seu nome para Promob, como estratégia de marketing.

FIGURA 13 – Organograma da Promob



Fonte: próprio autor

A U.D. Projetos é responsável pelo *software* de ambientação virtual, o Promob, e por demais produtos dependentes dele, como a ferramenta de customização, utilizada internamente e por alguns poucos clientes, chamada de Promob Catalog, e os plugins, como a ferramenta de renderização de projetos, Render Up.

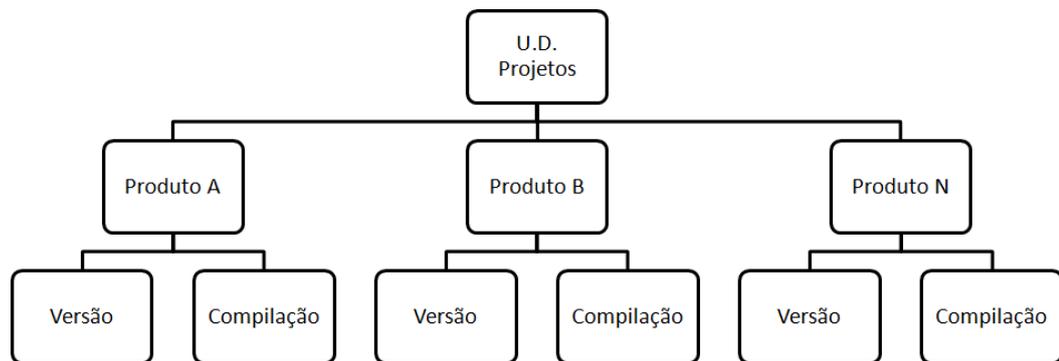
A U.D. Projetos é formada pelas equipes de Desenvolvimento, Testes, também chamada de Núcleo de Testes, e Documentação. As duas últimas gerenciam suas atividades independentemente e, prestam serviços para as demais unidades de desenvolvimento.

Os departamentos Relacionamento e Manutenção e Implantação, ambos pertencentes a área mercadológica, existentes na matriz e filiais, são clientes internos da área técnica, e realizam a customização do *software*. O resultado do trabalho desta equipe é uma biblioteca com o portfólio de produtos de um cliente, que juntamente com o núcleo do Promob, que possui ferramentas, compõem o produto de *software* Promob.

3.2 O PRODUTO PROMOB E SEU PROCESSO DE DESENVOLVIMENTO

O Promob é o produto mais representativo e com o maior número de usuários, são mais de 12 (doze) mil lojas de móveis em todo o país e por cerca de 6 (seis) mil profissionais das áreas de Marcenaria, Arquitetura e Design de Interiores. O cliente do Promob recebe dois tipos de atualização do produto, a compilação e a versão, conforme mostra a figura 14.

FIGURA 14 – Tipos de entrega do *software* Promob



Fonte: próprio autor

Uma versão possui melhorias nas funcionalidades existentes e novas funcionalidades. A versão do produto é atualizada anualmente. As alterações que ocorrem na versão, após seu lançamento, são as compilações, que basicamente são compostas por correções de erros, mas também pode conter melhorias consideradas urgentes. A compilação é um projeto curto, com entrega mensal. As novas necessidades ou situações não conformes são reportadas a U.D. Projetos através de solicitações (figura 15), que se torna um requisito de versão.

As solicitações são cadastradas no Portal de Solicitações, que é uma ferramenta interna, criada no *Microsoft SharePoint* e utilizada por todas as equipes da Promob, internas e externas, para solicitar algum serviço para outra área. A U.D. Projetos recebe solicitações dos atendentes de suporte, gerentes de relacionamento, equipe de treinamento e consultoria, colaboradores da área técnica das demais U.D. s e filiais.

O Portal de solicitações centraliza quase todos os requisitos que são implementados em uma entrega, mas não é a única forma de entrada. Requisitos de uma nova versão também podem ser recebidos na forma de um estudo de mercado, normalmente realizado pela área de marketing. Requisitos de uma compilação

podem ser tratados em uma reunião, principalmente quando se trata de integração entre produtos. Em nenhum dos casos, um documento ou ata de reunião é obrigatoriamente produzido, e o requisito poderá constar apenas no plano de projeto.

FIGURA 15 – Solicitação

ID	3.063		
Título	Intellicad fora do ar		
Departamento			
Serviço	Infraestrutura		
Anexo			
Interessados			
Avaliação	Viável		
Classificação			
Prioridade	1 - Alta		
Previsão de Entrega	24/9/2012		
Data de Conclusão	24/9/2012		
Discussão	<p>Camila Reiniciado o serviço. 24/9/2012 14:59:22</p> <p>Camila O servidor do Intellicad está fora do ar. 24/9/2012 14:39:34</p>		

Fonte: Portal de Solicitações Promob

Além de ser um canal de comunicação com a U.D., o portal é um gerenciador de requisitos para os projetos de desenvolvimento do Promob. Em paralelo ao gerenciamento realizado no portal, o plano de projeto (figura 16) permite a visão do andamento do projeto e gerenciamento dos recursos.

FIGURA 16 – Plano de projeto

	Nome da Tarefa	Andamento da Tare	Trabalho	Início	Término	% Concluído	Nomes dos recurso
	Compilação 5.34.5	●	924h	1/3/2012	15/6/2012	99%	
	Promob	●	604h	8/5/2012	5/6/2012	100%	
	Bugs	●	332h	8/5/2012	5/6/2012	100%	
	4297 - Ligar e Desligar camada Luz	●	8h	10/5/2012	10/5/2012	100%	
	4304 - erro na atualizacao do ambier	●	8h	11/5/2012	11/5/2012	100%	
	4012 - Ferramenta 30 40 50	●	8h	14/5/2012	14/5/2012	100%	
	1185 - 1722 - 4019 - 4370 - Correçã	●	40h	15/5/2012	21/5/2012	100%	

Fonte: Project Server

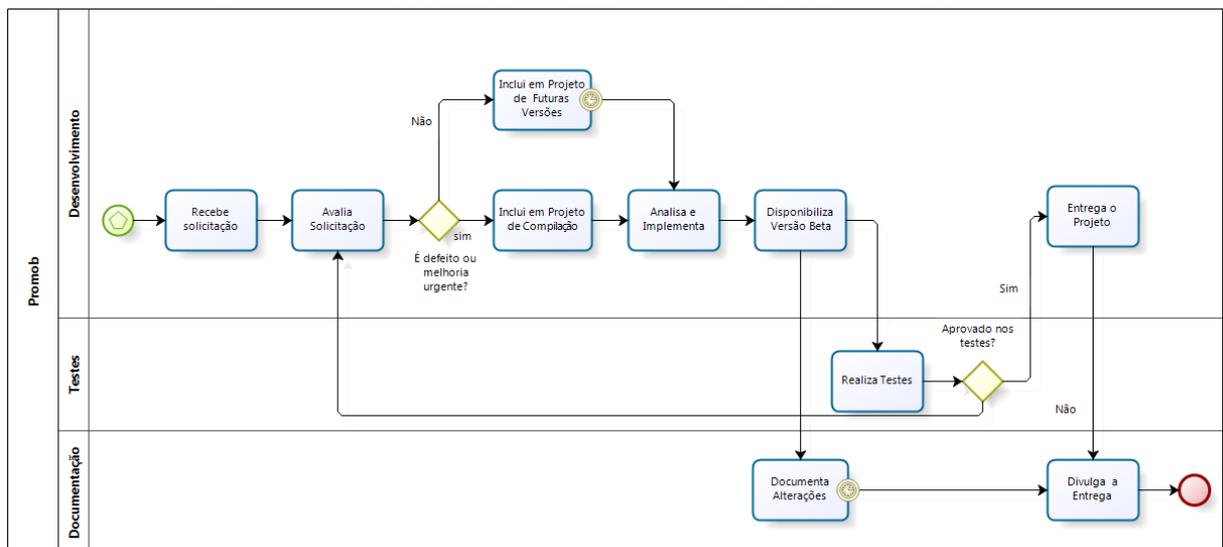
O plano de projeto, é uma lista de solicitações em um arquivo.project, que fica armazenado no MS Project Server, e lista os requisitos a serem atendidos em um

projeto de versão ou compilação, os recursos que executarão as tarefas e o registro de tempo para implementação da mesma.

Diariamente, o portal de solicitações é verificado, a fim de identificar os requisitos que serão atendidos. Os analistas de desenvolvimento possuem esta responsabilidade. Semanalmente, a versão beta do produto em desenvolvimento é disponibilizada para as equipes de documentações e testes, caracterizando o ciclo de desenvolvimento adotado como incremental.

O ciclo de desenvolvimento do Promob adota as etapas de planejamento, análise e implementação, documentação, testes e disponibilização. O processo de desenvolvimento é apresentado na figura 17, em BPMN - Business Process Modeling Notation (Notação de Modelagem de Processos de Negócio). O BPMN documentado encontra-se no anexo A.

FIGURA 17 – Processo de Desenvolvimento do Promob



Fonte: próprio autor

A solicitação cadastrada no Portal de Solicitações é recebida pelo responsável pelo produto, que é comunicado da existência da mesma através do envio de um e-mail, disparado pelo próprio portal (Recebe Solicitação). A mesma é avaliada por este responsável (Avalia Solicitação), que determina se a solicitação é viável ou não. Se viável, mas é uma melhoria não urgente ou uma sugestão de nova ferramenta, a solicitação será adicionada no projeto de futuras versões (Inclui em Projeto de Futuras Versões). Caso contrário, a solicitação descreve um erro ou uma nova necessidade de um cliente, que não pode aguardar até uma nova versão, esta solicitação entra em um projeto de compilação (Inclui em Projeto de Compilação). Se o plano do projeto da compilação ainda não existe, ele deve ser criado.

A equipe de desenvolvimento da U.D. Projetos é dividida por tipo de produto e, subdividida por tipo de entrega, compilação ou versão. Uma vez incluída em um projeto, a solicitação é atendida (Analisa e Implementa) pela equipe responsável. Conforme a complexidade da solicitação, se necessário, o gerente da U.D. ou um colega mais experiente faz o papel de analista de desenvolvimento. Esta análise ocorre de forma verbal e informal em conjunto com o desenvolvedor que implementará a mesma.

Ao concluir a tarefa, o desenvolvedor altera o estado da solicitação para concluído, no portal de solicitações. Em seguida, se reporta ao gerente da U.D., verbalmente ou via e-mail, para comunicar a conclusão da tarefa. O gerente então, atualiza o arquivo.project do projeto.

Uma versão beta do produto em desenvolvimento é gerada (Disponibiliza Versão Beta) com as solicitações atendidas até aquele momento. O responsável pela documentação dos requisitos utiliza a versão para documentar o requisito implementado (Documenta Alterações).

Na etapa de testes (Realiza Testes), casos de testes são criados e executados. Os requisitos não conformes retornam para a etapa de análise e implementação, como uma solicitação de correção, e repetem este caminho até que obtenham a aprovação ou que o gerente da U.D. opte por finalizar o projeto com o erro.

A finalização do projeto se dá com a entrega do projeto (Entrega o Projeto) disponibilizando o produto no diretório pré-definido e finalizando o projeto no *Project Server*.

Por último, ocorre a divulgação da entrega do produto de versão/compilação comunicado através do blog da U.D. Projetos (Divulga a Entrega).

As equipes de documentação e testes trabalham na versão beta gerada. Na etapa de documentação dois tipos de documentos são produzidos, os procedimentos direcionados ao usuário final e a *release notes*, divulgada na intranet da empresa, que informa o que foi implementado e é direcionada ao cliente interno.

Os testes são planejados a partir das solicitações, dos documentos produzidos pela equipe de documentação e, se necessário, de reuniões informais com o desenvolvedor responsável pela solicitação. Durante a execução dos testes é produzido o documento de registro de não conformidades, resultante dos testes

executados. Este tipo de solicitação é associado ao número da solicitação e classificada como:

- a) Teste unitário: quando a implementação não é encontrada no produto testável ou não é possível de executá-la.
- b) Erro: não há conformidade com a descrição da solicitação, com o documento de requisito implementado ou com a informação fornecida em reunião.
- c) Dúvida: a situação identificada parece ser um erro, mas não pode ser classificada como tal porque a documentação não deixa claro.
- d) Usabilidade: é um tipo de erro, que prejudica a usabilidade do *software*.
- e) Padrão: um padrão de interface acordado não foi seguido.
- f) Melhoria: a implementação possui o comportamento esperado. Entretanto, foi identificada uma melhoria para a função que gera um novo requisito. Se aprovada, a mesma pode ser atendida no projeto que a originou. Caso contrário, ela será transferida para o Portal de solicitações.

O documento de registro de não conformidades é uma planilha (figura 18), criada pelo testador e alterada pelo Gerente da U.D. e pelo testador. Normalmente, o mesmo desenvolvedor que atendeu a solicitação e gerou as não conformidades deve solucioná-las. Quando todos os requisitos forem aprovados a *Release candidate* é disponibilizada e é realizado o teste de integração.

FIGURA 18 – Solicitações de não conformidades

Núcleo de Testes							
Cadastro da solicitação							
Versão testada	Fase	ID Req	Nº s	Tipo Solicitação	Detalhes	Anexo	Desenvolvedor criação
Promob_5.34.6_04.07.12_14.20	Progressão	801	801.1	TU	Prateleiras com inserção Default estão sendo duplicadas conforme o número de vãos. Fernando já está ciente da situação.	S:\Nucleo de Testes\Projetos 2012\Compilacao 2012\34.6\Anexos\801.1.jpg	Fernando
Promob_5.34.6_04.07.12_14.20	Progressão	Ao adicionar objetos na impressão eles já ficam selecionados	2	Defeito	Ao inserir uma viewport em um documento de impressão, o retângulo cinza permanece sendo inserida por completo, o que causa um comportamento estranho se o cursor é movimentado após a inserção. Veja a imagem anexa.	S:\Nucleo de Testes\Projetos 2012\Compilacao 2012\34.6\Anexos\2.jpg	Samuel

Fonte: próprio autor

No projeto de versão, é acrescentado a execução de testes beta, onde clientes são selecionados e convidados a utilizar a versão beta para validar as alterações realizadas. A versão beta é atualizada mensalmente durante o período de testes beta da versão.

Na subseção a seguir se buscou definir o que é qualidade para o *software* ao qual um plano de qualidade será proposto, o Promob.

3.3 QUALIDADE PARA A EQUIPE PROMOB

Pressman (2006) recomenda fortemente, como primeiro passo em direção a implantação da Garantia da Qualidade que seja estabelecido o consenso sobre o conceito de qualidade. Buscou-se o envolvimento da empresa, e também dos clientes para esta definição. Para isso foram utilizadas duas pesquisas, cujo objetivo era definir o que é qualidade para o *software* Promob.

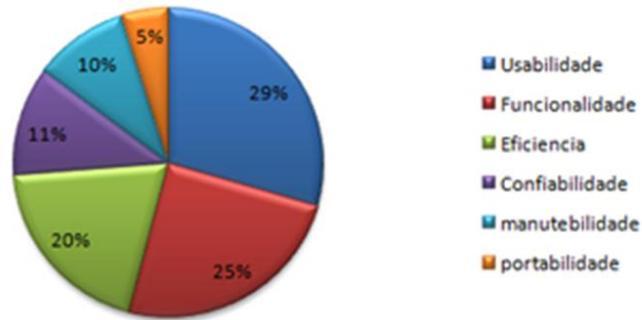
A pesquisa que envolveu a equipe da empresa, de diferentes níveis, gerenciais e operacionais, diretamente envolvidos com o processo de desenvolvimento do Promob, foi uma pesquisa quantitativa. Este tipo de pesquisa é realizada com questionários pré-elaborados que admitam respostas alternativas, cujos resultados são apresentados de modo numérico, permitindo uma avaliação quantitativa dos dados.

Para o questionário da pesquisa aplicada, utilizou-se o modelo de qualidade interna definido na NBR ISO/IEC 9126. Criada no formulário googleDocs (anexo B), a pesquisa foi enviada para 23 colaboradores convidados, 8 deste com cargo de nível gerencial e 15 com cargo de nível operacional. O formulário da pesquisa solicitava que fosse atribuída uma nota para cada uma das características e subcaracterísticas de qualidade estabelecidas pela NBR ISO/IEC 9126. As características: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade foram priorizadas, considerando a relevância da mesma para o produto Promob.

A pesquisa foi respondida por 67% dos convidados a respondê-la. Para analisar as respostas, identificar o equilíbrio e representatividade dos resultados, foram atribuídos pesos, utilizando o método do inverso da variância, ou seja, quanto maior a variabilidade, menor a participação do estudo na conclusão final (Matarazzo, 2003).

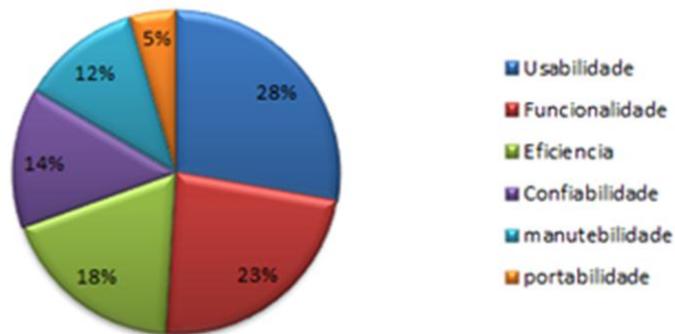
Na visão dos participantes que ocupam cargos operacionais, o resultado foi o representado na figura 19. A característica mais relevante foi usabilidade, seguida da funcionalidade. Eficiência ocupa a terceira posição, seguida de confiabilidade, manutenibilidade e portabilidade.

Na visão da gerência, também a característica mais relevante é a usabilidade, seguida da funcionalidade, depois eficiência, confiabilidade, manutenibilidade e portabilidade, conforme demonstra a figura 20.

FIGURA 19 – Gráfico de *resultado* operacional

Fonte: próprio autor

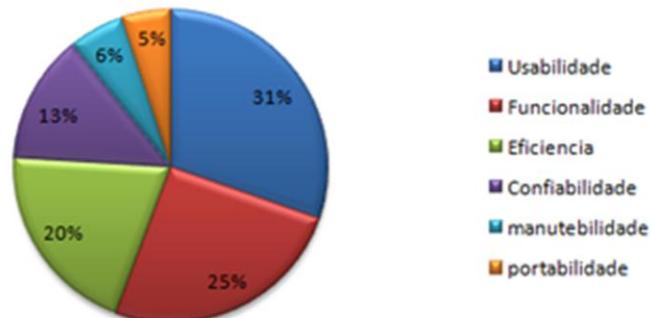
FIGURA 20 – Gráfico de resultado gerencial



Fonte: próprio autor

Observa-se que a visão operacional e gerencial estão alinhadas e não divergem quanto a priorização das características mais relevantes para o Promob. No ponto de vista de ambos, usabilidade, funcionalidade e eficiência ocupam as primeiras 3 (três) posições e concentram 76% da relevância, entre as seis características de qualidade de produto de *software* apresentadas (figura 21).

FIGURA 21 – Gráfico de resultado total



Fonte: próprio autor

3.4 QUALIDADE PARA O CLIENTE PROMOB

Para conhecer a visão do usuário, foi utilizado dados de uma pesquisa, realizada a cada 2 anos, aplicada por uma empresa terceirizada. A pesquisa, chamada Pesquisa de Satisfação tem foco comercial e o objetivo de avaliar a satisfação dos clientes quanto aos produtos e serviços oferecidos pela Promob.

Aplicada pela primeira vez no ano de 2010, a pesquisa avaliava apenas o produto Promob. Ela é segmentada em blocos de perguntas com mesmo objetivo: bloco de atendimento, bloco de serviços, bloco de políticas comerciais e bloco de produto. Este último apresenta as questões e resultados da tabela 1.

TABELA 1 - Pesquisa de Satisfação

Pergunta	Nível de Satisfação	Conceito
1 - Facilidade de instalação e configuração do <i>software</i>	81,10%	M. Bom
2 - Facilidade de uso (interface do sistema)	79,10%	Bom
3 - Facilidade de registrar e gerenciar a(s) licença(s) do <i>software</i>	78,70%	Bom
4 - Confiabilidade (Estabilidade e segurança do <i>software</i>)	77,40%	Bom
5 - Lançamentos e novidades da versão	74,10%	Bom
6 - Eficácia da documentação (as informações disponíveis no ícone de ajuda do <i>software</i> e na Base de Conhecimento lhe auxiliam na solução de dúvidas?)	71,90%	Bom
7 - Velocidade (durante a utilização)	68,10%	Regular

A coluna Nível de Satisfação é obtida através da média convertida em uma escala de 0 a 100% e o conceito calculado com base no resultado obtido, no nível de satisfação: maior que 90% corresponde a ótimo, entre 80% e 90% corresponde a muito bom, entre 70% e 80% corresponde a bom; entre 60% e 70% corresponde a regular e menor que 60% é considerado ruim.

As perguntas 1, 2, 3 e 6 são relacionadas à característica de usabilidade. A pergunta 4 com a característica confiabilidade, 5 com a funcionalidade e 7 com a eficiência, conforme subcaracterísticas determinadas pela NBR ISO/IEC 9126. Os usuários pontuaram e um dos atributos de usabilidade, operacionalidade, presente na questão “Facilidade de instalação e configuração do *software*” recebeu conceito

“muito bom”. Porém na média, usabilidade, confiabilidade e funcionalidade obtiveram satisfação de 77,7%, 77,4% e 71,1%, respectivamente, o que classificou estas características com conceito “bom”. Eficiência é a característica que merece mais atenção no ponto de vista do usuário, obtendo conceito “regular”.

3.5 CONSIDERAÇÕES

A pesquisa realizada contou com o apoio e envolvimento da equipe do Promob, despertando interesse e curiosidade sobre os resultados da mesma. Quanto ao resultado obtido, mostra o alinhamento entre a preocupação da empresa com a melhoria das características, usabilidade e funcionalidade e a avaliação dada pelos clientes na Pesquisa de Satisfação. Usabilidade e funcionalidade obtiveram conceitos “muito bom” e “bom”, respectivamente. Sem minimizar a relevância destas duas características, a terceira colocada, eficiência, deve receber no mínimo igual preocupação para satisfazer os usuários do Promob.

A característica Usabilidade e suas subcaracterísticas inteligibilidade, apreensibilidade, operacionalidade e atratividade tratam da capacidade do produto de *software* de ser compreendido, aprendido, operado e de ser atraente ao usuário, quando usado sob condições especificadas. A presença desta característica nas primeiras posições tem alguns motivos conhecidos, como ser este um dos mais forte argumento de venda do Promob diante de seus concorrentes. Pode ainda ter influencia um recente e forte movimento da empresa em relação a esta característica.

A característica funcionalidade, e suas subcaracterísticas adequação, acurácia, interoperabilidade e segurança de acesso, relacionadas à capacidade do produto de *software* prover funções que atendam às necessidades explícitas e implícitas. As funções, também chamadas de ferramentas, é além de diferencial competitivo, motivo de grande preocupação a cada planejamento de nova versão para atender as diferentes necessidades de seus clientes e manter a competitividade entre eles.

Já a característica Eficiência, que evidencia o relacionamento entre o nível de desempenho do *software* e a quantidade de recursos usados, sob condições estabelecidas, é um desafio maior, primeiramente porque o Promob é um *software* de processamento gráfico. Logo envolve o desempenho e o uso de recursos da

máquina. Pode ser afetada não apenas pelo núcleo do Promob, desenvolvido pela U.D. Projetos, mas também pela forma que a customização é realizada, esta é influenciada pela estrutura de produto do cliente.

A confiabilidade ocupa a quarta posição. Embora seja extremamente relevante, a maturidade, uma das subcaracterísticas desta característica conforme a IEC/ISO 9126, conquistada ao longo dos 14 anos de mercado que o produto Promob possui, permitem que atualmente ele disponibilize de mecanismos de tolerância a falhas e recuperabilidade, outras duas subcaracterísticas da característica confiabilidade.

A manutenibilidade determina a capacidade do produto de *software* de ser modificado, desde correções, melhorias ou adaptações do *software*. Como todas as solicitações de alteração são de alguma forma atendidas e o custo das alterações não é controlado e o impacto que uma alteração pode gerar, como uma funcionalidade crítica parar de funcionar não é visto como um problema de manutenibilidade, e sim apenas como um erro de implementação. Também não existe nenhum controle efetivo sobre este tipo de defeito, até este momento.

A portabilidade ter sido considerada a menos relevante está diretamente relacionado às políticas de licenciamento de *software* adotadas pela empresa, que limita, restringe e estabelece regras e condições para instalações, uso, troca de computadores entre outros.

As pesquisas estabelecem as características mais relevantes e a prioridade das mesmas para o produto. Desta forma, a garantia da qualidade do Promob deve focar em usabilidade, funcionalidade e eficiência, principalmente.

4. PROPOSTA DO PLANO DE GARANTIA DA QUALIDADE PROMOB

O SQAP - Plano de Garantia da Qualidade - é um artefato que visa a formalização do processo de Garantia da Qualidade na organização. O documento define os padrões organizacionais apropriados a um determinado produto e processo de desenvolvimento. Estabelece as qualidades desejadas para o produto a ser desenvolvido e, como essas qualidades devem ser avaliadas. Para tal, fornece um roteiro que contempla o conjunto de atividades da estratégia, como os processos de verificação, validação e revisão. Estabelecendo a visão sobre todos os recursos e esforços a serem utilizados para garantir a qualidade do *software*.

Para atender a estes itens, a IEEE 730 apresenta uma estrutura de 16 seções e referencia outras normas específicas para cada um dos documentos solicitados nas seções. Entretanto, a avaliação dos documentos e especificações da Promob não é contemplada por este estudo e proposta.

As seções da IEEE 730 serão utilizadas como referencia para estabelecer o SQAP para o Promob, adotando 15 (quinze) seções, das 16 (dezesesseis), sugeridas na norma. No processo da U.D. Projetos nem todos os documentos requeridos pela IEEE 730 são produzidos. Os documentos de maior relevância foram criados para possibilitar a aplicação desta proposta, os demais ficam como sugestão.

No SQAP serão definidos importantes e relevantes itens do processo de garantia da qualidade, conforme definição das normas CMMI, MPS.BR e ISO/IEC 12207 que descrevem metas e práticas, abordadas no capítulo 2, que resultam em padrões, atividades e técnicas direcionadas a produzir um *software* com qualidade.

Por ser um projeto contínuo, o Promob permite um único documento SPQA padrão para os seus diferentes tipos de projetos, versão e compilação, abordados no capítulo 3. Alguns itens de determinadas seções somente poderão ser aplicadas a versão, em consequência de ser a compilação um projeto realizado em curto prazo.

As seções deste capítulo especificam cada seção do Plano de Garantia da Qualidade.

4.1 PROPÓSITOS

A primeira seção descreve o objetivo específico e o escopo de aplicação do SQAP, definindo o(s) produto(s) de *software* contemplados pelo plano e o uso

pretendido para este *software*. Bem como, as etapas do ciclo de vida do *software* contempladas para atingir os requisitos e o nível de qualidade desejada para o *software*.

Com o objetivo de ser um guia para a equipe da U.D. Projetos, direcionando os esforços para atingir os níveis de qualidade desejados para produto Promob, revelados no capítulo 3. Para tal, é estabelecida a sistematização das atividades de SQA no processo de desenvolvimento do *software*.

O plano contempla o processo de desenvolvimento do núcleo do *software* Promob, que abrange as ferramentas que possibilitam criar um projeto de layout de ambiente e as ferramentas utilizadas para a customização. Os dois tipos de projetos desenvolvidos pela U.D. Projetos, versão e compilação, são cobertos pelo plano. Entretanto, algumas medições somente serão realizadas no processo de versão. A compilação é um projeto mensal e, basicamente, composto por requisitos de correção. O curto ciclo de vida do projeto inviabiliza a avaliação de métricas mais complexas, por exemplo, métricas relacionadas a usabilidade.

As atividades de SQA estão presentes em todas as etapas do processo de desenvolvimento do *software* Promob: planejamento, análise e implementação, documentação, testes e disponibilização.

4.2 DOCUMENTOS DE REFERÊNCIA

Esta seção fornece a lista dos documentos que são referenciados durante as seções do SQAP. Para cada documento deve ser indicado a versão, a data do mesmo e o repositório onde ele está disponível. São documentos de referência.

- a) Processo de desenvolvimento da U.D. Projetos: o processo existente foi modelado em BPMN para ser analisado, avaliado e ter o plano de garantia da qualidade proposto, especificado no capítulo 2.
- b) Pesquisa de Atributos de Qualidade: a pesquisa realizada a partir das qualidades determinadas na ISO 9126 permitiu identificar o que era qualidade para o *software* Promob, identificando e estabelecendo consenso sobre os atributos de qualidade desejáveis para o produto, apresentada no capítulo 3.
- c) Plano de testes: documento utilizado como guia para definir os testes e especificar os casos de testes a serem executados para o Promob, que foi revisado e adequado com os resultados obtidos através deste estudo.

- d) IEEE 730 - Padrão para Plano de Garantia de Qualidade de *software*: norma utilizada para estabelecer este documento, o Plano de Garantia de Qualidade de *software*.

4.3 GESTÃO

A seção 3 determina as tarefas do plano de garantia da qualidade e as responsabilidades atribuídas a cada envolvido, que influencia e/ou controla a qualidade do *software*. Esta seção está dividida em organização, atividades, responsáveis e recursos de SQA:

4.3.1 Organização

O Núcleo de Testes é a equipe que tem como objetivo a qualidade de *software*. É reconhecida pela empresa como um departamento distinto, mas está subordinado a U.D. Projetos, que é responsável pelo produto Promob como um todo, do negócio ao desenvolvimento.

Nesta subseção, cada cargo/papel desempenhado na U.D. deve ser descrito, juntamente com a responsabilidade a este atribuída. Deve ser apresentado também o responsável pela preparação e manutenção do SQAP. Deve estar explícito o nível de autonomia do(s) responsável(is) por avaliar e monitorar a qualidade do *software*, bem como verificar resolução de problemas identificados no processo.

O SQAP, sendo uma atividade cujo foco é a qualidade, deve ser de responsabilidade da equipe de Testes. A equipe possui familiaridade com o tema e conhecimento para absorver as atividades de SQA devido ao perfil dos integrantes desta equipe que está muito próximo ao desejado: colaboradores com conhecimento em desenvolvimento e engenharia de *software*, com bom relacionamento e facilidade em comunicar-se, conhecem todos os produtos, a integração entre eles, bem como os processos para construí-los, além de ter contato com clientes na execução de testes beta (Lazzari, 2009).

Além do Núcleo de Testes, a U.D. Projetos é formada pelo seu gerente, gerente do projeto, este apenas como papel, analistas de desenvolvimento, desenvolvedores, documentador, testador. Cada cargo/papel possui responsabilidades para com a Garantia da Qualidade:

- a) O gerente da U.D. é responsável por:
- pela aprovação do planejamento das ações apresentadas pelo Núcleo de Testes;
 - pelo planejamento das ações de qualidade do processo;
 - por providenciar a resolução de não conformidades do processo;
- b) O gerente de produto é responsável por providenciar a resolução de não conformidades do produto.
- c) O(s) documentador(es) são responsáveis por traduzir os requisitos técnicos em uma linguagem que possa ser compreendida pelas demais equipes da Promob e por clientes;
- d) O(s) analista(s) de desenvolvimento e desenvolvedor(es) são responsáveis:
- Identificar soluções para os requisitos do *software*;
 - Implementar as soluções propostas nas versões/compilações do *software*;
- e) O(s) analista(s) de teste e testador(es) são responsáveis:
- pelo planejamento e execução de qualquer ação de qualidade de produto U.D.;
 - pela identificação e reporte de não conformidades de produto e processo;
 - pelo acompanhamento de resolução de não conformidades;
 - pela avaliação da resolução de não conformidades;
 - pelo desenvolvimento e manutenção do Plano de Garantia da Qualidade do Promob.

4.3.2 Tarefas

A subseção Tarefas descreve as etapas do ciclo de vida do *software*, determinando as atividades de SQA a serem executadas, suas entradas e saídas. As tarefas, representadas na tabela 2, determinam os pontos de controle estabelecidos no processo de desenvolvimento do produto e são alinhadas com o cronograma do projeto.

TABELA 2 - Plano de SQA

Etapa	Principais Produtos	Atividades SQA
Planejamento do projeto	Plano de produto	Revisar plano de produto
Análise e Implementação	Solicitação	Revisar análise de solicitação
	Executável testável	Revisar código fonte
Documentação	Documento requisito implementado	Revisar documento de requisito implementado
	Documento de divulgação	Revisar o documento de divulgação
Testes	Casos de Testes	Revisar os Casos de Testes
Disponibilização	Produto de software final	Revisar o produto de software final
Processo de SQA	Processo de SQA	Avaliar a execução do processo

a) São tarefas do SQA para a etapa de Planejamento:

- Revisar plano de produto: esta tarefa consiste em identificar se todas as solicitações do projeto foram relacionadas no projeto, para garantir que as mesmas serão testadas, documentadas e divulgadas. A revisão ocorre através da comparação do documento do Project produzido com as solicitações que possui estado de concluída no portal de solicitações.

b) São tarefas do SQA para a etapa Análise e Implementação:

- Revisar análise de solicitação: nesta atividade, executada a cada solicitação atendida, ou seja, requisito implementado, visa que o desenvolvedor revise a solução dada e a execução da mesma.
- Revisar código fonte: nesta revisão o produto de trabalho da implementação deve ser avaliado quanto a sua aderência a práticas adotadas pela U.D.

c) São tarefas do SQA para a etapa Documentação:

- Revisar documento de requisito implementado: este documento explica a necessidade e a solução dada para as alterações realizadas para a entrega. Deve ser criado um para cada solicitação listada no plano de produto.
- Revisar documento de entrega: realizado na conclusão de cada projeto, este documento deve conter todas as solicitações listadas no plano de projeto, bem como o link para a documentação realizada para a mesma, sua revisão deve verificar a aderência a estes.

d) São tarefas do SQA para a etapa de Testes:

- Revisar casos de testes: os casos de testes são produzidos antes de cada etapa de testes e devem contemplar as solicitações atendidas no executável testável disponibilizado, sua revisão deve verificar a aderência aos atributos de qualidade do Promob.

e) São tarefas do SQA para a etapa de Disponibilização:

- Revisar produto final de *software*: realizada na conclusão do projeto, a revisão garante que o executável do produto pode ser disponibilizado. A disponibilização requer o produto ofuscado, compilado e disponibilizado no diretório pré-definido.

f) Ao final do projeto, a atividade de auditoria do processo de SQA deve ser realizada. Esta consiste em verificar se todas as atividades de SQA foram realizadas, produzindo os documentos devidos com qualidade.

4.3.3 Funções e responsabilidades

As responsabilidades de SQA devem ser descritas nesta seção. Entretanto, toda a equipe da U.D. Projetos: gerente, analistas, desenvolvedores, documentador e testadores são responsáveis pela garantia da Qualidade, cada qual pelo produto do qual é autor ou responsável.

4.3.4 Recursos para SQA

Esta deve fornecer as estimativas de recursos e os custos a serem despendidos na garantia da qualidade e controle de tarefas de qualidade. O custo da obtenção da qualidade é referente ao esforço a ser realizado para alcançá-la.

O custo da qualidade de *software* envolve o custo da conformidade e da não conformidade. Bartié (2002) divide os custos da conformidade em duas categorias: custos da detecção de defeitos, relacionados às atividades que visam detectar problemas nos produtos de *software* gerados no ciclo de desenvolvimento, e custos de prevenção de defeitos, relacionados a todas as atividades que buscam redução de falhas nas diversas etapas do processo. Os custos da não conformidade é o gerado por todas as atividades ligadas ao esforço de correções de falhas ocorridas durante o processo de desenvolvimento.

Segundo estudos de Myers (1989) quanto ao tempo para identificar e corrigir um erro, conhecida como Regra de 10, significa que a cada etapa que um erro passa sem ser identificado o seu custo de correção multiplica 10 vezes.

Atualmente a U.D. Projetos trabalha com um custo de até 15% do tempo de desenvolvimento do projeto para a etapa de testes. Percentual que deve ser mantido. Entretanto, este percentual agora não será mais utilizado apenas pelo Núcleo de Testes, mas sim por toda a equipe, durante todo o ciclo de desenvolvimento do produto.

4.4 DOCUMENTAÇÃO

Esta seção possui duas importantes funções. A primeira é identificar a documentação que rege o desenvolvimento, verificação e validação, uso e manutenção do *software*. A segunda é identificar como os documentos devem ser verificados quanto à adequação.

O Promob é mais antigo produto da empresa e possui a equipe mais madura. Entretanto, a criação de documentos de padrões, normas, processos, procedimentos entre outros, não é uma prática e todos utilizam do próprio conhecimento tácito e dos demais elementos da equipe.

Para esta seção, a norma estabelece requisitos mínimos para garantir que a implementação do *software* satisfaz os requisitos técnicos. É recomendado que o conhecimento tácito identificado fosse formalizado e assim torne-se explícito, o que permitiria que esta seção fosse completada.

4.4.1 Documentos para V&V, uso e manutenção do *software*

4.4.1.1 Descrição de requisitos de *software*

O documento de requisitos de *software* deve especificar os requisitos para um produto de *software* específico, programa ou conjunto de programas que certas funções em um ambiente específico. Pode ser escrito pelo fornecedor, cliente ou por ambos. Deve abordar as questões básicas de funcionalidade, interfaces, de desempenho, atributos e restrições de design imposta na execução.

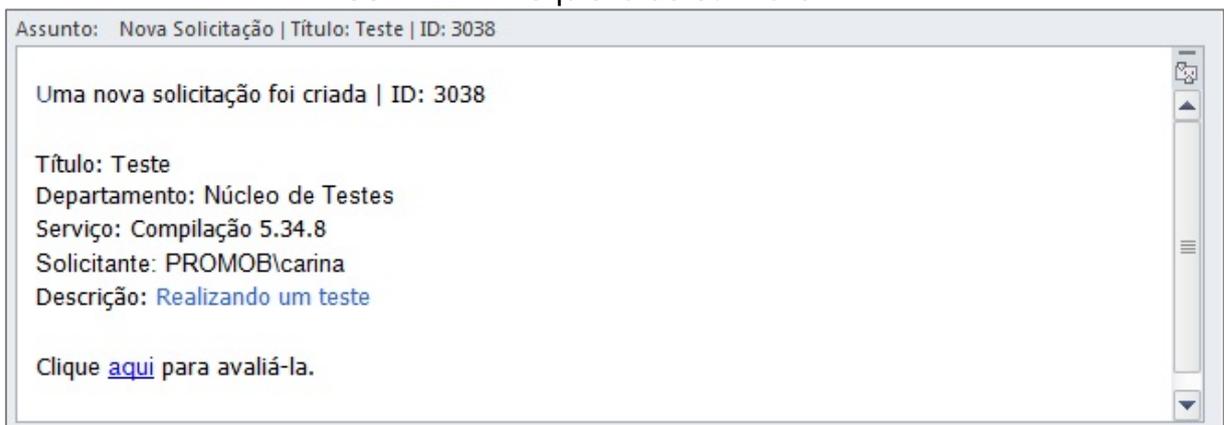
O documento que a U.D. Projetos usa como requisito é a solicitação (figura 22), do Portal de solicitações apresentado no capítulo 4. Descritas pelo cliente e

transcritas por uma equipe da Promob, como a Central de Atendimento, as solicitações reportam somente informações referente a necessidade ou problema. Das demais formas de entrada de solicitações/requisitos, verbalmente ou estudo de mercado, sugere-se que também sejam convertidas em solicitação, mesmo que criada pelo solicitado, criando assim documentos para as mesmas e disponibilizando de forma centralizada.

4.4.1.2 Descrição do projeto *software*

Este documento deve descrever a forma como o *software* será estruturado de forma a satisfazer os requisitos da descrição de requisitos de *software*. Ele deve descrever os componentes e subcomponentes do projeto de *software*, incluindo bases de dados e interfaces, em resumo a arquitetura do *software*. Sugere-se que esta estrutura para o Promob seja documentada.

FIGURA 22 – Requisito de *software*



Fonte: Portal de solicitações

4.4.1.3 Plano de V & V (verificação e validação)

Processos de verificação e de validação são utilizados para determinar se os produtos de software desenvolvidos estão conforme os requisitos, para que cumpram o uso pretendido e as expectativas dos usuários. O plano de V&V documenta as tarefas de verificação e validação, podendo ser um ou dois documentos.

Os planos foram construídos em um documento (tabela 3), espelhados no processo validação das normas de qualidade de processo estudadas, conforme fases do processo de desenvolvimento do produto Promob.

4.4.1.4 Relatório de resultados de V&V

O relatório de resultados de verificação e validação deve apresentar os resultados das atividades de *software* realizadas de acordo com o plano de V&V (figura 23). Este documento não é gerado atualmente, ele passa a existir a partir da aplicação deste plano.

A publicação do relatório deve ocorrer na página do Núcleo de Testes para que todos os interessados tenham acesso ao resultado de SQA obtido.

TABELA 3 - Plano de V & V

Plano	Entrega	Nível	Estratégias	Abordagem	Fase	Categoria	Método para obter CT	Responsáveis
Validação	Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Testes (Testes de Unidade)	Funcionalidade	Decomposição de requisitos	Desenvolvedor
	Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Testes (Testes de Unidade)	Usabilidade	Decomposição de requisitos	Desenvolvedor
	Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Testes (Testes de Unidade)	Segurança	Decomposição de requisitos	Desenvolvedor
	Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Aceitação)	Funcionalidade	Decomposição de requisitos	Analista Testes/Testador
	Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Aceitação)	Usabilidade	Decomposição de requisitos	Analista Testes/Testador
	Versão	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Aceitação)	Segurança	Decomposição de requisitos	Analista Testes/Testador
	Versão	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Sistema)	Volume	Decomposição de requisitos	Analista Testes/Testador
	Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Integração)	Compatibilidade	Decomposição de requisitos	Analista Testes/Testador
	Versão	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Sistema)	Performance	Decomposição de requisitos	Analista Testes/Testador
	Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Sistema)	Confiabilidade e Disponibilidade	Decomposição de requisitos	Analista Testes/Testador
	Versão	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Sistema)	Recuperação	Decomposição de requisitos	Analista Testes/Testador
Verificação	Versão e Compilação	Alto	NA	NA	Análise e Implementação	Verificação do documento de solicitação	NA	Desenvolvedor
	Versão e Compilação	Alto	NA	NA	Análise e Implementação	Verificação do código fonte	NA	Desenvolvedor

Fonte: próprio autor

FIGURA 23 – Modelo de relatório V&V

Selecione o projeto:		Compilação 4
Período de desenvolvimento		
De:		
Até:		
REQUISITOS ATENDIDOS		
Prioridade	Quantidade	
Alta	1	
Média	1	
Baixa	1	
Não avaliado	0	
Total	4	
Etapa: PLANEJAMENTO		
Verificações realizadas	1	
Verificações não realizadas	1	
Conformidades	1	
Não conformidades	0	
Etapa: ANÁLISE E IMPLEMENTAÇÃO		
Verificações realizadas		
Verificações não realizadas		
Conformidades		
Não conformidades		
Etapa: DOCUMENTAÇÃO		
Verificações realizadas		
Verificações não realizadas		
Conformidades		
Não conformidades		
Etapa: DISPONIBILIZAÇÃO		
Verificações realizadas		
Verificações não realizadas		
Conformidades		
Não conformidades		
Etapa: TESTES		
Verificações realizadas		
Verificações não realizadas		
Conformidades		
Não conformidades		
SEVERIDADE DOS ERROS		
Prioridade	Quantidade	
Defeito	3	
Melhoria	2	
TU	1	
Usabilidade	1	

Fonte: próprio autor

4.4.1.5 Documentação do usuário

A documentação do usuário orienta os usuários na instalação, operação, gerenciamento e manutenção dos produtos de *software*. Deve ser descrito as entradas de controle de dados, seqüências de entrada, limitações e todas as informações essenciais para o produto de *software*. Mensagens de erro devem ser identificadas e descritas, bem como as ações corretivas para corrigir os erros que causam as mensagens de erro.

A documentação produzida para o usuário na Promob é depositada em uma base *online*, chamada de base de conhecimento (figura 24). Esta documentação deve especificar e descrever os dados necessários para uso do *software*. São dois os tipo de documentação. A primeira escrita em formato de dica de utilização, ou seja, através de um exemplo aplicado deve fornecer a seqüência de entradas, as opções e limitações do programa e outras atividades ou itens necessários para a execução bem sucedida das funcionalidades do *software*. A segunda são os registros de mensagens de erro esperadas que devem ser descritos, bem como os procedimentos corretivos que se aplicam. Os requisitos cuja implementação não tem interação direta com o usuário não tem necessidade de ser documentado.

FIGURA 24 – Base de conhecimento



Fonte: Base de conhecimento Promob

4.4.1.5 Plano de gerenciamento de configuração do *software*

Este plano deve documentar o gerenciamento de configuração do *software*, estabelecendo as atividades a serem realizadas, os responsáveis pelas tarefas específicas, além de definir os métodos e instalações utilizados para manter, conservar, proteger e versões de documentos controlados e artefatos relacionados do *software* identificados durante todas as fases do ciclo de vida do *software*.

Na etapa de disponibilização, o SVN – Subversion – que é um serviço de repositório de dados, é usado para controle de versões de sistemas. Nele são centralizadas todas as operações de leitura e escrita de arquivos codificados, oferecendo recursos de histórico de alterações, resgate de arquivos a partir de um determinado período e linhas de alterações paralelas, que permite gerenciar separadamente mais de uma versão do mesmo arquivo. O uso desta solução é bem importante para possibilitar a equipe trabalhar em uma versão, produzindo compilações, enquanto desenvolve uma nova versão.

Utilizada em conjunto com o Subversion, uma aplicação própria, o Forrest, cujo propósito é de manter um mapeamento entre as estruturas do cliente, onde as alterações são feitas, e do servidor SVN, onde os arquivos são armazenados. Nesse ambiente, toda alteração realizada é mapeada seguindo os padrões internos para o servidor, e baixada para alteração seguindo o mesmo padrão de mapeamento. Este mesmo recurso é utilizado para isolar a versão/compilação que é testada, a cada ciclo incremental. Estas são formalmente documentadas.

Os documentos gerados pelas atividades de SQA, que devem ser armazenados, devem ser armazenados no servidor do Núcleo de testes, conforme políticas de acesso existentes. Um diretório, chamado SQA, deve ser criado e dentro deste um novo diretório cujo nome é o ano corrente (2012), este deverá conter um diretório para cada produto, nomeado com o nome do mesmo (Promob), seguindo padrão já utilizado neste servidor. Cada novo projeto, um novo diretório filho do produto deve ser criado, com o nome do projeto (compilacao2012) e neste serão depositados os arquivos de SQA, que não estiverem hospedados no servidor do Sharepoint.

4.4.2 Produtos a serem verificados

Os documentos a serem verificados devem ser identificados e incluir os critérios e a identificar a técnica pela qual devem ser verificados, quanto a adequação aos requisitos. São documentos a serem verificados:

- a) Plano de Projeto: arquivo.project (figura 25) que fornece a lista de solicitações que serão atendidas em um projeto de compilação ou versão. Permite gerenciar os recursos e o tempo durante o projeto.

FIGURA 25 – Plano de projeto

Nome da Tarefa	Andamento da Taref	Trabalho	Início	Término	% Concluído	Nomes dos recurso
Compilação 5.34.5	●	924h	1/3/2012	15/6/2012	99%	
Promob	●	604h	8/5/2012	5/6/2012	100%	
Bugs	●	332h	8/5/2012	5/6/2012	100%	
4297 - Ligar e Desligar camada Luz	●	8h	10/5/2012	10/5/2012	100%	Samuel Gubert -
4304 - erro na atualizacao do ambier	●	8h	11/5/2012	11/5/2012	100%	Samuel Gubert -
4012 - Ferramenta 30 40 50	●	8h	14/5/2012	14/5/2012	100%	Samuel Gubert -
1185 - 1722 - 4019 - 4370 - Correçã	●	40h	15/5/2012	21/5/2012	100%	Samuel Gubert -

Fonte: MS Project Server

- b) Código-fonte: solução codificada (figura 26) computacionalmente correspondente a necessidade apresentada no requisito.

FIGURA 26 – Código fonte

```

ObjetoViewport.cs
ProMOB.Impressao.ObjetoViewport

private void AplicarEscala()
{
    AplicarEscala(ViewCtrl, true);
}

private void AplicarEscala(ViewCtrl view, bool atualizarAssociados)
{
    if (!ViewEhValida(view)) return;
    if (view.ViewInterfaceEstatica.PtVista.Perspectiva)
        return;

    view.PermiteZoom = EscalaOrtogonal.Livre;

    if (!EscalaOrtogonal.Livre)
    {
        float largura = RetanguloViewPort.Width;
        float altura = RetanguloViewPort.Height;
    }
}

```

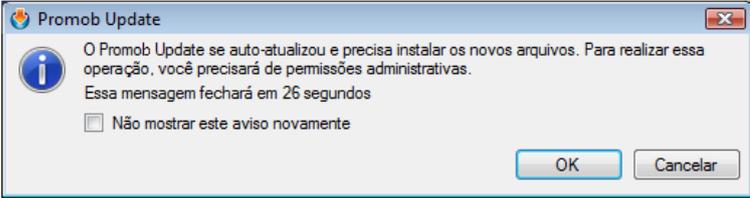
Fonte: MS Visual Studio

- c) Documento de requisito implementado: documento que registra a necessidade e a solução dada para a solicitação (figura 27). Informa para quais distribuições se aplicam as implementações realizadas, descreve o que a solução permite e não permite realizar e como usá-la. Para toda solicitação, ou requisito este documento é criado.

FIGURA 27 – Documento de requisito implementado

Quando ocorre?

Ocorre ao iniciar a atualização do Promob.



Por que ocorre?

Ocorre porque o Promob Update foi atualizado e precisa instalar os arquivos da atualização. Para que os arquivos da atualização sejam instalados, é necessário que o usuário logado no computador possua permissões administrativas.

Como resolver?

1. Na mensagem que é exibida, clique em **OK** e inicie a atualização do Promob novamente;
2. Caso a situação persista, executar o Promob como **Administrador** e após, execute novamente o processo de atualização.

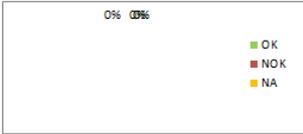
Links relacionados:

1. [Como atualizar o Promob?](#)
2. [Como executar o Promob como administrador?](#)

Fonte: Base de conhecimento Promob

d) Casos de Testes: documento que descreve situações de testes específicos para a solução implementada (figura 28).

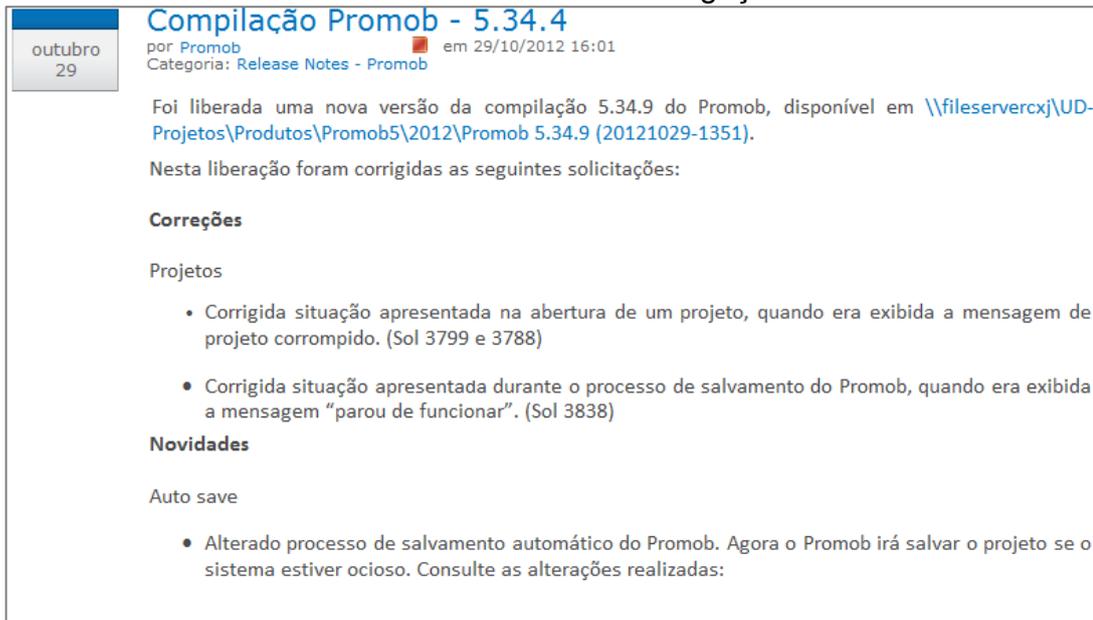
FIGURA 28 – Casos de Testes

CASOS DE TESTES					
INFORMAÇÕES DO PROJETO					
Nome do Projeto					
Nº do Projeto					
Versão testada					
RESULTADOS					
Aprovado	0		Casos de teste		
Reprovado	0		Casos de teste criados	0	
NA	0		Casos de teste executados	0	
			Casos de teste aprovados	0	
			Casos de teste não-aprovados	0	
			Casos de teste não executados	0	
Correções					
Caso	Nº Req	Situação	Resultado esperado	Status	Observação
CT					
Melhorias					
Caso	Nº Req	Situação	Resultado esperado	Status	Observação
CT					

Fonte: Documento modelo de Casos de Testes Promob

- e) Documento de divulgação: documento que comunica da disponibilização fornece a lista de solicitações atendidas e o link para a base de conhecimento, onde os documentos de requisito implementado estão hospedados (figura 29).
- f) Produto final de *software*: executável compilado do *software* Promob.

FIGURA 29 – Documento de divulgação



Fonte: Base de conhecimento Promob

4.5 NORMAS, PRÁTICAS, CONVENÇÕES E MÉTRICAS

Esta seção identifica as normas, as práticas e as convenções a serem utilizados no projeto de *software* contemplado pelo SQAP. As medições também são estabelecidas nesta seção.

A U.D. adota os padrões de programação OO e cria seus códigos em língua inglesa, com a orientação de que ao alterar códigos mais antigos, seja mantido o padrão do mesmo. O padrão de interface é parcialmente documentado, a criação de mensagens para o usuário deve seguir o padrão existente e documentado. O padrão de telas é seguindo através da observação de telas já existentes. Recomenda-se a documentação destas práticas e padrões.

As métricas que possibilitam analisar o processo de desenvolvimento de *software* estão relacionadas a detecção de defeitos, conforme tabela 4.

TABELA 4 - Métricas do Processo de Desenvolvimento

Seleção da Métrica	Descrição da métrica	Fórmula
Retrabalho	Retrabalho	retrabalho = (Número de requisitos implementados/tempo de implementação)/ (Número de requisitos implementados/tempo de correção).
Prioridade dos defeitos	Defeito de impacto baixo	Impacto = (número de solicitações do tipo "Defeito" e de priorização "Média" e priorização "Baixa")/ número total de defeitos.
	Defeito de impacto médio	Impacto = (número de solicitações do tipo "Defeito" e de priorização "Alto")/ número total de defeitos.
	Defeito de impacto alto	Impacto = (número de solicitações do tipo "Defeito" e de priorização "Urgente")/ número total de defeitos. Urgente: situações onde o software é finalizado durante a execução ou de alguma forma impede o cliente de dar seguimento a suas atividades através do software, tal como impossibilitar gerar qualquer tipo de saída.

Métricas sobre os dados planejamento e execução dos testes e de eficiência das atividades de verificação e da validação estão diretamente associados ao processo de SQA. Na tabela 5 são apresentadas as métricas propostas para avaliar o processo de Garantia da Qualidade proposto.

TABELA 5 - Métricas do Processo de SQA

Processo	Seleção da Métrica	Descrição da métrica	Propósito	Fórmula
Processo de SQA	Distribuição dos erros por etapa	Planejamento	Erros identificados na etapa de planejamento	Erro = número de ocorrência "de não conformidades" na etapa de planejamento/número total de requisitos
		Análise e implementação	Erros identificados na etapa de análise e implementação	Erro = número de ocorrência "de não conformidades" na etapa de análise e implementação/número total de requisitos
		Documentação	Erros identificados na etapa de documentação	Erro = número de ocorrência "de não conformidades" na etapa de documentação/número total de requisitos
		Testes	Erros identificados na etapa de testes	Erro = número de ocorrência "de não conformidades" na etapa de testes/número total de requisitos
		Disponibilização	Erros identificados na etapa de disponibilização	Erro = número de ocorrência "de não conformidades" na etapa de disponibilização/número total de requisitos
	Testes	Severidade dos erros	Classificação dos erros conforme tipo de situação reportada	Erro de funcionalidade = número de ocorrência de erros na funcionalidade/número total de erros
				Erro de Padrão
				Erro de Usabilidade
				Erro TU
				Erro de Padrão = número de ocorrência de erros por falta de padrão/número total de erros
Erro de Usabilidade = número de ocorrência de erros de usabilidade/número total de erro				
Erro TU = número de ocorrência de erros de TU/número total de erros				

As métricas de produto avaliam as características desejadas para o *software*. Para a seleção de métricas considerou-se as medições indicadas na NBR ISO/IEC 9126-2, comparada a capacidade do *software* de ser medida e a realidade da

empresa, que introduz o conceito de garantia da qualidade e medições. Bem como, o resultado da priorização das características (tabela 6), realizada através da atribuição de pesos, conforme respostas dadas a pesquisa aplicada com a equipe Promob, apresentada no capítulo 3, a mesma que determinou as características de qualidade para o *software* Promob.

TABELA 6 - Priorização das características de qualidade

Características de Qualidade	Subcaracterísticas de Qualidade	Prioridade da característica	Prioridade da subcaracterística
Usabilidade	Compreensibilidade	3	4
Usabilidade	Aprendizabilidade	3	3
Usabilidade	Operacionalidade	3	2
Usabilidade	Atratividade	3	1
Funcionalidade	Adequação	2	4
Funcionalidade	Segurança de acesso	2	3
Funcionalidade	Interoperabilidade	2	2
Funcionalidade	Acurácia	2	1
Eficiência	Comportamento de medidas de tempo	1	1
Eficiência	Medidas de utilização de recursos	1	2

As métricas apresentadas nas tabelas 7, 8 e 9 devem ser aplicadas para avaliar o produto nas suas três características desejadas: usabilidade, eficiência e funcionalidade. Esta avaliação exige maior tempo e por ser um projeto de compilação mensal, as mesmas somente devem ser aplicadas a projetos de versão, normalmente anuais.

TABELA 7 - Métricas de funcionalidade

Subcaracterísticas de Qualidade	Seleção da Métrica	Fórmula
Adequação	Plenitude da implementação	$X = 1 - A / B$ A = Número de funções em falta detectada em avaliação B = Número de funções descrito no especificações de requisitos
	Cobertura da implementação	$X = 1 - A / B$ A = Número de funções implementadas incorretamente ou funções em falta detectada na avaliação B = Número de funções descrito no especificações de
Acurácia	Precisão	$X = A / T$ A = Número de resultados encontrados pelos usuários com o nível de precisão diferente exigido T = tempo de operação
Segurança	Prevenção na corrupção de dados	a) $X = A / N$ A = Número de vezes que projetos ou documentos do projeto corrompeu N = Número total de usuários ativos do Promob

TABELA 8 - Métricas de usabilidade

Subcaracterísticas de Qualidade	Seleção da Métrica	Fórmula
Compreensibilidade	Manifestação eficácia	$X = A / B$ A = Número de funções operado com sucesso B = Número de demonstrações/tutoriais acessados
Aprendizabilidade	Facilidade de aprendizagem da função	T = tempo médio necessário para aprender a usar um funcionar corretamente
	Acessibilidade da base de conhecimento	$X = A / B$ A = Procedimentos pesquisados na base de conhecimento, sem retorno B = Total de número de procedimentos pesquisados
	Eficácia da base de conhecimento	$X = A / B$ A = Número de chamados abertos com pedidos de ajuda para as funcionalidades do produto, concluídos com êxito, com apenas uma interação B = Total de chamados abertos com pedidos de ajuda para as funcionalidades do produto
Operacionalidade	Facilidade de correção de erros	$T = T_c - T_s$ Tc = Tempo de completar correção de especificados erros de tipo de tarefa executada Ts = Tempo de correção a partir de determinado erros de tipo de tarefa executada
	Compreensibilidade de mensagem de erro	$X = A / UOT$ A = número de vezes que o usuário faz uma pausa por um longo período de tempo ou sucessivamente, e repetidamente falhar na mesma operação, devido à falta de compreensão da mensagem. UOT = tempo de operação durante período de observação
	Auto explicatividade de mensagens de erro	$X = A / B$ A = Número de condições de erro para o qual o usuário propõe a ação de recuperação correta B = Número de condições de erro testado

Muitas das métricas de usabilidade são possíveis de serem medidas através da monitoria de usuários operando o Promob, reforçando a sugestão de adotar testes alfa.

TABELA 9 - Métricas de eficiência

Subcaracterísticas de Qualidade	Seleção da Métrica	Fórmula
Comportamento de medidas de tempo	Tempo de resposta	$T = (\text{tempo de ganhar o resultado}) - (\text{Tempo de entrada de comando concluído})$
Comportamento de medidas de tempo	Tempo de espera	T = Tempo entre usuário terminar ficando resultados de saída e solicitar acabamento usuário

4.6 REVISÕES

Esta seção define quais as revisões do *software* a serem realizadas, como devem ser aplicadas e o momento em que devem ser executadas, relacionando com o cronograma do projeto de *software*.

Técnicas estáticas e dinâmicas permitem verificar e validar os produtos de *software* produzidos. Das técnicas estáticas a revisão de *software* é mais apropriada, e entre seus tipos, a revisão técnica, juntamente com a Auditoria, são os tipos de revisões mais citadas e recomendadas para a garantia da qualidade. A Auditoria por possibilitar avaliar processo e produto com foco principal na avaliação das atividades realizadas e as revisões técnicas por avaliarem a conformidade do produto em relação às especificações e planos definidos. Ambas são técnicas formais, exigem papéis bem definidos e treinamento, o que as tornam mais efetivas na detecção de erros e defeitos em tempo de corrigi-los e ainda aperfeiçoar a qualidade, porém mais difíceis de implementar.

As inspeções e *walkthrough*, esta última menos formal, colaboram com a mudança de comportamento necessária neste processo e podem ser utilizadas para detecção de não conformidades nos produtos de trabalho intermediários. A aplicação de ambas se dá de forma semelhante iniciando após a disponibilização do produto a ser revisado, especialistas examinam o produto de *software* e a documentação pertinente, identificando e registrando os defeitos encontrados.

Walkthrough tem como objetivo educar os envolvidos da organização sobre a necessidade da utilização de revisões sobre os produtos de trabalho e auxilia na detecção de várias anomalias, como problemas de modularidade no projeto e no código, além de especificações que não podem ser testadas na prática. Sendo recomendada como uma etapa inicial para organizações que ainda não trabalham a revisão de seus produtos de *software*. A Inspeção é uma revisão formal, planejada, com papéis bem definidos, reuniões formais e aplicada com o auxílio de *checklists*.

Diferentes técnicas podem ser utilizadas para avaliar os produtos nas diferentes fases de desenvolvimento. Para os produtos do processo de desenvolvimento do Promob será utilizado *walkthrough* para avaliar produtos de *software* e Auditoria para avaliar processo, ambas planejadas com *checklists*.

4.7 TESTES

Esta seção deve identificar todos os testes não incluídos na verificação de *software* e um plano de validação para o *software* coberto pela SQAP. Deve indicar os métodos a serem utilizados. Se um documento de plano de testes existe, deve ser referenciado.

Recentemente elaborados, os Planos de Verificação e de Validação contemplam todos os testes realizados para o Promob. O documento de Plano de Testes não é um documento utilizado. O plano de Validação elaborado é apresentado na tabela 10.

TABELA 10 - Plano de Validação

Entrega	Nível	Estratégias	Abordagem	Fase	Categoria	Método para obter CT	Responsáveis
Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Teste de Unidade	Funcionalidade	Decomposição de requisitos	Desenvolvedor
Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Teste de Unidade	Usabilidade	Decomposição de requisitos	Desenvolvedor
Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Teste de Unidade	Segurança	Decomposição de requisitos	Desenvolvedor
Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Teste de Aceitação	Funcionalidade	Decomposição de requisitos	Analista de Testes/Testador
Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Teste de Aceitação	Usabilidade	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	Baseado em requisitos	Teste de Aceitação	Segurança	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	Baseado em requisitos	Teste de Sistema	Volume	Decomposição de requisitos	Analista de Testes/Testador
Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Teste de Integração	Compatibilidade	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	Baseado em requisitos	Teste de Sistema	Performance	Decomposição de requisitos	Analista de Testes/Testador
Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Teste de Sistema	Confiabilidade e Disponibilidade	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	Baseado em requisitos	Teste de Sistema	Recuperação	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	Baseado em requisitos	Teste de Sistema	Contingência	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	NA	Validação do Sistema	Estrutural	Grupos do sistema	Clientes convidados
Versão	Alto	Caixa-preta	NA	Validação do Sistema	Beta-teste	ND	Clientes convidados

4.8 RELATÓRIO DE NÃO CONFORMIDADES E AÇÕES CORRETIVAS

Esta seção descreve as práticas e procedimentos a serem seguidos para a comunicação, acompanhamento e resolução de problemas nos produtos de *software*, no processo de desenvolvimento de *software* e no processo de manutenção, na Promob chamado de compilação.

Uma planilha MS Office Excel é utilizada como Relatório de não conformidades. No entanto esta é vulnerável, qualquer colaborador da equipe pode alterar a mesma, comprometendo a integridade dos dados. No entanto, o mesmo Portal de Solicitações pode ser utilizado para o reporte de não conformidades identificadas nas revisões de SQA. A utilização do portal além de garantir a integridade dos dados, facilita a interação entre desenvolvedor e testador e descentraliza a manutenção das solicitações, pois uma vez determinado que cada

desenvolvedor é responsável pela correção dos erros referentes a solicitação que atendeu, não é necessário que o gerenciamento destas solicitações seja realizado por um terceiro, atualmente um papel desempenhado pelo gerente da U.D. Projetos.

Uma solicitação (figura 30) passaria pelas fases: cadastro, atendimento e conclusão. Na inclusão, o solicitante seleciona o Departamento, no caso seria selecionado “Núcleo de Testes”, no campo Serviços se seleciona o projeto a que a mesma se refere, por exemplo, “Compilação (identificador_da_compilação)”. Os campos Título e Descrição são livres e permitem ao solicitante relatar a situação ou necessidade que deseja reportar. O responsável por atender a solicitação deve ser incluído como interessado e o mesmo recurso associado a solicitação no plano de projeto. O interessado recebe, automaticamente, um e-mail informando da inclusão da solicitação.

FIGURA 30 – Cadastro de Solicitação

A imagem mostra uma janela de software intitulada "Solicitações - Novo Item". No topo, há uma barra de ferramentas com ícones para Salvar, Fechar, Copiar, Recortar e Colar, além de botões para Confirmar e Área de Transferência. O formulário principal contém os seguintes campos:

Departamento	Núcleo de Testes
Serviço	Compilação 5.34.8
Título	Teste
Descrição	Realizando um teste
Interessados	Marcio - Promob ;
Anexos	Clique aqui para anexar um arquivo
	Padrão

Fonte: Portal de solicitações

No atendimento, o responsável providencia a correção e uma vez finalizada, edita a solicitação (figura 31) e altera o campo Avaliação para “viável” e o *status* da mesma para “concluída”. Caso a situação não possa ser reproduzida, mesmo com auxílio da equipe de testes, ou não possa ser atendida por limitação técnica, por exemplo, o campo *status* é alterado para “inviável” e neste caso a solicitação será automaticamente cancelada. Esta alteração de *status* dispara um e-mail para o

solicitante, no caso, o Núcleo de Testes reportando a alteração ocorrida na solicitação.

FIGURA 31 – Edição de Solicitação

Solicitação Atendida?	Sim
ID	3.038
Título	teste
Departamento	Núcleo de Testes
Serviço	Compilação 5.34.8
Anexo	 Clique aqui para anexar um arquivo
Interessados	Marcio - Promob ;  
Previsão de Entrega	
Data de Conclusão	24/9/2012
Discussão	<div style="border: 1px solid gray; padding: 5px;"> <p>Mauricio teste 24/9/2012 8:24:56</p> </div>

Fonte: Portal de solicitações

Se a situação foi corrigida, a solicitação é finalizada ao responder “sim” para a questão “Solicitação atendida”, senão ao responde “não” a mesma é reincluída e retorna para a etapa de atendimento.

4.9 FERRAMENTAS DE APOIO

Esta seção deve identificar as ferramentas de *software* especiais, técnicas e metodologias que suportam SQA, seu estado fins, e descrever o seu uso.

A Promob disponibiliza recursos, em sua maioria Microsoft, devido a uma parceria que possui com esta empresa. A comunicação entre departamentos na empresa é formalizada através de soluções implementadas no MS *Sharepoint* (utilizado para intranet, página de departamentos, *blogs*, portal de solicitações e para base de conhecimento). O Portal de Solicitações é utilizado para visualização dos requisitos do projeto (solicitações) e para cadastro de solicitações, resultante da verificação e validação de produtos de *software* e de Auditorias de SQA. A página do departamento Núcleo de Testes deve ser utilizada para apresentar o relatório de validação e verificação do plano. No blog ocorre o comunicado de disponibilização de um produto entregue. Na base de conhecimento são criados e publicados os

documentos de requisitos implementados. No MS Office Excel são criados os relatórios e casos de testes e o GoogleDocs será utilizado para criar os *checklists* das atividades de SQA.

4.10 CONTROLE DE MÍDIA

Esta seção trata de identificar o acesso ao produto de *software*, a armazenagem e a proteção de acesso não autorizado ou danos. Esta definição é de responsabilidade da área de infraestrutura da empresa que controla o acesso no servidor, sendo que cada departamento da empresa possui o seu, com restrições de acesso para os demais colaboradores (somente leitura), bem como a estrutura de *backup* interna e externa.

4.11 CONTROLE DE FORNECEDORES E CONTRATADOS

Esta seção deve indicar as disposições para assegurar que o *software* disponibilizado pelos fornecedores atende as exigências. Deve indicar os métodos que serão utilizados para assegurar que o *software* recebido está adequado e completo. Entretanto, o único fornecedor de *software* da empresa é a Microsoft e não cabe a Promob impor quaisquer exigências a eles.

O *software* Promob não possui contratados, não havendo disposições a apresentar nesta seção quanto a este item. Por estes dois motivos, o SQAP proposto não irá dispor desta seção.

4.12 REGISTROS, MANUTENÇÃO E RETENÇÃO

Esta seção deve identificar a documentação SQA que deve ser mantida, indicando os métodos e as instalações a serem usado para montar, arquivar, salvar e manter esta documentação, e designará o período de retenção.

Os documentos a serem armazenados são os *checklists*, as solicitações de não conformidades de SQA e métricas (tabela 11). Solicitações são armazenadas no servidor do *sharepoint*, por tempo indeterminado. Os demais documentos devem ser armazenados no servidor da U.D., também por tempo indeterminado.

TABELA 11 - Tabela de documentos a armazenar

Documento	Periodo	Diretório
Solicitações	Tempo indeterminado	Servidor Sharepoint
<i>Checklists</i> de revisão	Tempo indeterminado	Servidor da U.D. Projetos
Relatórios de V & V	Tempo indeterminado	Servidor da U.D. Projetos
Métricas	Tempo indeterminado	Servidor da U.D. Projetos

4.13 TREINAMENTO

Esta seção deve identificar as atividades de formação necessárias para atender às necessidades do SQAP. A apresentação dos conceitos de SQA e do Plano de Qualidade deve ser realizada, juntamente com a definição das atividades, cronograma e responsáveis. Preenchimento de *checklist*, comunicação e armazenamento também devem ser difundidos.

Para o treinamento sugere-se a prática já adotada na U.D., por e-mail o assunto é introduzido, uma reunião em sala reservada quando o número de pessoas é maior que três, ou então no próprio local de trabalho. Para colaboradores novos, o treinamento deve ser realizado quanto este tiver contato, pela primeira vez, com o processo de SQA. No processo atual, o analista que apadrinhar o colaborador é responsável por identificar as necessidades de treinamentos internos, bem como o momento em que ele deve ocorrer e agendar com o responsável, no caso o analista de testes que assume o papel de analista de SQA.

4.14 GESTÃO DE RISCOS

Esta seção especifica os métodos e procedimentos empregados para identificar, avaliar, monitorar e controlar áreas de risco que ocorram durante o ciclo de vida do *software* coberto pelo SQAP.

Os projetos da U.D. Projetos não são formalmente gerenciados quanto aos riscos. O Núcleo de testes disponibilizava do levantamento de riscos para execução de testes. Este documento serviu como base para identificar os riscos relacionados a SQA, apresentados na tabela 12.

TABELA 12 - Tabela de riscos

Risco	Probabilidade	Gravidade	Contingência
Atraso ou não entrega dos requisitos para as atividades de testes.	Média	Média	Contatar com superior imediato.
Levantamento equivocado ou distorcido dos requisitos.	Alta	Alta	Contatar com gerente do projeto responsável.
Falta de conhecimento no contexto da implementação.	Média	Alta	Verificar documentos disponíveis
			Solicitar reunião com envolvidos com o negócio.
Afastamento temporário do analista/testador.	Baixa	Alta	Ajuste do cronograma do projeto.
Especificações funcionais insuficientes ou inexistentes	Média	Alta	Contatar com gerente responsável pelo projeto
			Solicitar reunião com envolvidos com o negócio.
			Ajuste do cronograma.
Não ter contato com os interessados	Média	Média	Contatar com superior imediato.
Atraso na entrega do projeto	Alta	Baixa	Contatar com gerente do projeto.
			Ajuste do cronograma.

4.15 GLOSSÁRIO

Nesta seção os termos usados no SPQA devem ser listados com breve explicação ou definição.

4.16 PROCEDIMENTO DE CONTROLE MUDANÇA DO SQAP E HISTÓRICO

Esta seção deve citar os procedimentos para alteração e/ou atualização do SQAP, mantendo o histórico de tais modificações. Dados, conforme tabela 13, devem ser disponibilizados no documento do plano para que esta seção seja atendida, disponibilizar as colunas: Data, onde deve ser informada a data em que o documento foi alterado, o número de versão deve ser alterado e preenchido na coluna Versão, o campo Descrição deve ser informado o que motivou a alteração e o Responsável pela modificação.

TABELA 13 - Controle de Mudanças

Data	Versão	Descrição	Responsável

4.17 CONSIDERAÇÕES DA PROPOSTA

O SPQA, Produto do Processo de Garantia da Qualidade, é o guia da garantia da qualidade dos produtos e do processo de desenvolvimento de *software*, que atua de maneira pró ativa. Através do Plano de SQA (quadro 32), definindo as atividades, os documentos que devem ser revisados, através de qual técnica, por quem e em qual momento para que seja possível a identificação de erros, em cada uma das etapas do processo pela qual passa um requisito, evitando que erros se tornem defeitos. O documento proposto encontra-se no anexo J.

FIGURA 32 – Plano de SQA

Etapa	Principais Produtos	Atividades SQA	Técnica	Crítérios	Fase	Documento de Apoio	Responsáveis
Planejamento do projeto	Plano de produto	Revisar plano de produto	Walkthrough	Verificar se não há solicitações pendentes no portal de solicitações Verificar se as tarefas foram incluídas no plano de projeto e se recurso foi atribuído Verificar se o Plano de Projeto está atualizado Verificar se o Plano de Projeto foi publicado	A cada release candidate disponibilizada	Checklist do plano de produto	Gerente da UD
Análise e Implementação	Solicitação	Revisar análise de solicitação	Walkthrough	Verificar se todos os relacionamentos e dependências com a alteração a ser realizada foram previstos Avaliar o impacto da alteração a ser realizada Verificar se o projeto da nova versão/compilação foi baixado para o computador do desenvolvedor antes de realizar alguma alteração Verificar se a interface segue os padrões acordados entre a equipe Verificar se nenhuma mensagem apresentada pelo sistema ao usuário final possui erros gramaticais Verificar se todas as mensagens e alteração de interfaces foram traduzidas para os idiomas espanhol e inglês Verificar se exceções críticas foram tratadas	A cada solicitação implementada	Checklist da análise	Desenvolvedor responsável
		Revisar código fonte	Walkthrough	Verificar se existe legibilidade no código implementado Verificar se sincronizou com projetos do SVN antes de comitar alterações Verificar se a alteração foi realizada em todas as versões em andamento	A cada solicitação implementada	Checklist da implementação	Desenvolvedor responsável
Documentação	Documento requisito implementado	Revisar documento de requisito implementado	Walkthrough	Verificar se todos as solicitações, que exigiam documentação, foram documentados Verificar se documento segue os padrões Verificar disponibilidade do documento	A cada solicitação implementada	Checklist de requisito implementado	Documentador
	Documento de divulgação	Revisar o documento de divulgação	Revisão	Verificar se todas as solicitações atendidas estão documentadas Verificar se o documento está formatado conforme padrão Verificar se o documento está disponível para consulta	Após finalização da compilação/ versão (produto final)	Checklist de divulgação	Documentador
Testes	Casos de Testes	Revisar os Casos de Testes	Walkthrough	Verificar se os casos de testes estão claros e objetivamente definidos. Verificar se os casos de testes contemplam os tipos de testes obrigatórios. Verificar se os tipos de testes são focados nos atributos de qualidade do software. Verificar se os casos de testes possuem todos os cenários possíveis para identificação de falhas.	A cada executável beta gerado	Checklist de Casos de Testes	Analista de testes
Disponibilização	Produto final	Revisar o produto final	Walkthrough	Verificar se todos os projetos de código-fonte alterados foram comitados. Verificar se o executável testável está disponível no repositório pré estabelecido Verificar se executável foi gerado corretamente, com todos os arquivos necessários	Após finalização da compilação/ versão (produto final)	Checklist de entrega	Responsável pela gerencia de configuração
Processo de desenvolvimento	Processo da UD-Projetos	Avaliar a execução do processo	Auditoria	Verificar se todas as solicitações atendidas foram concluídas Verificar status de execução dos casos de teste Verifica se a base de conhecimento foi atualizada	Após entrega compilação/ versão (produto final)	Checklist de SQA	Responsável por SQA

Fonte: próprio autor

A análise do processo de desenvolvimento, o qual não deveria ser alterado pela proposta, as definições de qualidade, através das pesquisas, e de ter os documentos mais necessários sido desenvolvidos, o plano de garantia da qualidade foi proposto. Adequando o modelo estrutural da norma IEEE 730 para a realidade da empresa e de seu produto Promob.

O plano de verificação definido, tem como referência os resultados estabelecidos em SG1 do CMMI e VER 1, VER 2 e VER 3 do MPS.BR. No entanto, a dificuldade em elaborá-lo evidencia a deficiência de documentação gerada pelo processo de desenvolvimento da U.D. Projetos. Para o plano de validação, a empresa obtinha um documento chamado de Plano de Testes, este nem era um plano de testes nem atendia ao plano de verificação, mas permitiu extrair o plano de validação, depois de revisado e adequado.

5. APLICAÇÃO DO PLANO DE QUALIDADE

No período de realização deste trabalho, o ciclo de desenvolvimento do produto Promob, produto para o qual o SQAP foi proposto, encontra-se em fase de manutenção, ou seja, projetos de compilação estão sendo desenvolvidos. Embora, uma nova versão esteja sendo projetada e comece a ser desenvolvida. Este cenário direcionou a escolha do projeto ao qual o plano de qualidade seria aplicado e medições realizadas. Juntamente com o gerente da UD, foi selecionado o período e a compilação que adotaria o SQAP.

Entre os meses de setembro e outubro de 2012, a equipe responsável pelo desenvolvimento do projeto de compilação trabalhou utilizando o Plano de Qualidade como guia. O projeto de compilação em questão foi composto por 54 requisitos de correção, uma de melhoria para ferramentas de customização e 12 de melhorias solicitadas por clientes. E será chamada, ao longo deste capítulo, de compilação 4.

A aplicação do SQAP no projeto da compilação 4 transcorreu sem grandes surpresas. A aplicação, as mudanças necessárias no SQAP e os resultados obtidos serão apresentados nas próximas seções.

5.1 O SQAP NO PROCESSO DE COMPILAÇÃO

A primeira atividade relacionada a aplicação do SQAP foi realizada uma semana antes de iniciar o projeto da compilação 4. Através de *e-mail*, forma adotada pela U.D. para comunicar e registrar alterações na forma de trabalho ou compartilhar conhecimento, as alterações que ocorreriam para o projeto de compilação a ser iniciado foram apresentadas a equipe envolvida, bem como o motivo pelo qual ocorreria. Na mesma semana, a reunião mensal, na qual a equipe participa junto a seu gerente, o mesmo solicitou a participação de todos os envolvidos com o projeto.

Dias antes do início do projeto da compilação 4, uma reunião com os analista, os mesmos que participaram do desenvolvimento dos *checklists*, foi marcada. A pauta desta era a aplicação dos *checklists*. Durante a reunião, a forma de preenchimento dos mesmos foi discutida.

No dia que o projeto foi iniciado, uma reunião foi realizada com cada desenvolvedor participante do projeto, individualmente e no próprio local de trabalho, apresentando o *checklist* que seria aplicado na etapa com o qual estava envolvido e

esclarecendo as possíveis dúvidas. Enfatizou-se que a avaliação tinha foco no processo e no produto, e que o resultado dependia em muito das informações que seriam fornecidas nos *checklists*. Destacou-se que não seria um problema responder negativamente um item do *checklist*. Por exemplo, para a pergunta 1 (um) do *checklist* de análise que questionava “A solução proposta, além de solucionar o problema apresentado, atende a necessidade do cliente?” responder “Não conheço a necessidade do cliente” não seria um problema. Dever-se-ia responder com sinceridade e posteriormente buscar conhecer a necessidade, analisar a solução do requisito e novamente responder o *checklist* para a solicitação em questão.

As tarefas foram realizadas conforme determinava o Plano de SQA:

a) Tarefa: Revisar Análise:

A primeira tarefa de SQA aplicada foi a de revisão de análise, através do *checklist* de análise (anexo C). Aplicada pelo desenvolvedor, que também é analista, do requisito que atende.

b) Tarefa: Revisar código fonte

Realizada na sequência da revisão da análise, também pelo desenvolvedor que atende o requisito, através do *checklist* de implementação (anexo D).

c) Revisar Documento de Divulgação

Seguindo o processo de desenvolvimento da U.D. Projetos, a cada versão beta disponibilizada as solicitações de melhoria eram documentadas. Para estes o *checklist* de requisito implementado (anexo E) foi aplicado pelo documentador.

d) Revisar Casos de Testes

Na etapa de testes, os casos de testes criados foram verificados e validados, conforme *checklist* de casos de testes (anexo F), pelo analista de testes ou testador que os criou, para depois serem aplicados. Os requisitos reprovados pelos testes retornaram à etapa de desenvolvimento para correção. Para este um novo ciclo é iniciado, até que todos os erros sejam corrigidos ou que o gerente da U.D. opte por finalizar o projeto com os erros identificados, como ocorreu na compilação 4. Neste projeto foram executados três ciclos, no primeiro todos os casos de testes são executados, nos demais as solicitações de testes e o(s) caso(s) de testes nos quais os erros foram identificados são reexecutados.

e) Revisar Plano de Produto

Ao finalizar o projeto, que ocorreu dois dias antes da entrega do mesmo, tarefa Revisar Plano de Produto foi realizada, através da aplicação do *checklist* do plano de produto, pelo gerente da U.D. A proposta previa a aplicação deste a cada disponibilização, inclusive das versões beta, porém não ocorreu desta forma.

No período entre a proposta e a aplicação do SQAP em um projeto, esta atividade foi, em parte, automatizada. Ela era necessidade de diferentes pessoas, envolvidas e em diferentes etapas. A aplicação criada possibilita percorrer um projeto, pré-selecionado, no Project Server e retorna as solicitações concluídas por data em que foram adicionadas ao projeto, permitindo ao usuário identificar se o plano está atualizado bem quais os itens adicionados.

Neste novo cenário, aplicar o *checklist* do plano de produto (anexo G) torna-se necessário apenas ao finalizar o projeto, para garantir que todas as solicitações atendidas estão listadas no plano de projeto.

f) Revisar Documento de Divulgação

O documento de divulgação, construído em paralelo a documentação de requisito implementado, pode ser finalizado após o fechamento do projeto. O mesmo foi verificado através do *checklist* de entrega (anexo H), pelo responsável pela atividade de documentação.

g) Revisar Produto Final de *software*

O produto final de *software* gerado passou pela revisão aplicada pelo responsável pela gerencia de configuração, com o *checklist* de entrega (anexo I) proposto.

h) Auditar a execução do processo de SQA

Após a entrega, a auditoria do processo foi realizada, conforme *Checklist* de Auditoria (anexo I).

O preenchimento do *checklist* de revisão do plano de produto foi respondido no momento adequado e executado com sucesso, conforme verificou a auditoria, realizada pelo analista de testes e responsável pelo processo de SQA. Todas as solicitações atendidas na compilação foram listadas no plano de produto; o recurso a qual cada solicitação foi atribuída estava correspondente ao respectivo desenvolvedor que a implementou; as solicitações atendidas foram concluídas no

portal de solicitações e nenhuma solicitação urgente, recebida no período da compilação e possível de ser solucionada, deixou de ser atendida.

O resultado da etapa de análise e implementação foi o que gerou mais expectativa, visto que envolve um maior número de pessoas, era a equipe que poderia oferecer maior resistência a adoção dos *checklists*, e centraliza uma das atividades de maior impacto, a análise de desenvolvimento. Nesta etapa, a auditoria identificou que 20 requisitos não foram avaliados contra os *checklists*, sob a justificativa de esquecimento, caracterizando não conformidades com o processo de SQA.

Nas tarefas de documentação, documento de requisito implementado e documento de entrega estão centralizadas na única pessoa, que responde e executa as atividades da área, o que facilita para que *templates* sejam seguidos. Não foi identificado qualquer inconformidade nas revisões desta etapa.

Na etapa de testes são avaliados os casos de testes, referente ao planejamento dos testes. A auditoria identificou 12 não conformidades de SQA, que corresponde a 12 requisitos que não tiveram casos de testes planejados, devido a características do mesmo, simples e pontual como, por exemplo, a tradução de um campo de texto. Na análise de não conformidades, a auditoria identificou que para cada Caso de Teste reprovado, havia pelo menos uma solicitação de correção reportada a equipe de desenvolvimento, que é o mínimo esperado.

Com sucesso foi realizada a verificação do produto final de *software*, através do *checklist* de entrega, sendo esta a última verificação do produto de *software* antes de ser entregue.

5.2 MUDANÇA NO SQAP

As seções que compõem o SQAP possuem diferentes funções, seções 1 e 2 são de orientação, determinando o escopo e os resultados esperados, bem como o referencial teórico do plano. As seções 3, 4 e 6 são de aplicação e estabelecem as tarefas de SQA, o como e o quando para as mesmas. A seção 5 trata das medições. Da seção 7 a 14 são descritas as seções de apoio.

As seções de orientação determinam os objetivos do documento Plano de Garantia da Qualidade, o escopo do produto e do processo, estabelece as características de qualidade desejáveis para o *software* e define as etapas de

atuação do SQAP no ciclo de desenvolvimento. A seção 2 cita os documentos de referência do plano. Estas seções não sofreram alteração.

A partir da seção 3 são apresentadas as atividades de aplicação, que são as tarefas de SQA que devem ser executadas durante as etapas do processo, as funções e as responsabilidades dos envolvidos. Na seção 4 são apresentados os documentos a serem revisados, pelas tarefas definidas na seção 3, e a documentação referencial para o desenvolvimento do produto e processos de verificação, validação, uso e manutenção do *software*. A seção 6 apresenta o como e quando realizar as revisões dos produtos de *software* do projeto. Para estas seções são criados os *checklists*, de verificação e validação, e um deles sofreu alteração.

No primeiro dia de aplicação deste, surgiu uma solicitação de melhoria, que se referia ao momento de preenchimento do *checklist*. Inicialmente, foi solicitado que os desenvolvedores ao entenderem que concluíram a solicitação e antes de submetê-la, respondessem o *checklist*. Entretanto, uma das questões solicitava: “A alteração realizada foi submetida (comitada)?”. Logo, a resposta sempre seria negativa. Para solucionar o impasse foi acordado que o preenchimento do *checklist* ocorria sincronizado com o momento de conclusão da solicitação, ou seja, o preenchimento do *checklist* ocorreria imediatamente antes de submeter a alteração para o servidor. O *checklist* de implementação foi o único a ser alterado. Os demais foram aplicados sem qualquer dificuldade, dúvida ou sugestão de melhoria.

Outra alteração ocorreu quanto a forma de preenchimento dos *checklists*. Ao contrário do inicialmente proposto, *checklists* criados no MS Office Excel, foi adotado formulários no GoogleDocs, a alteração ocorreu ao em reunião, juntamente com os analistas, antes de iniciar o projeto da compilação 4, identificar que nem todos tinham acesso local ao *software* MS Office Excel. A empresa disponibiliza este, localmente, apenas para algumas pessoas, as que utilizam a ferramenta com muita frequência, e as demais utilizam o recurso *online* e compartilhado, este possibilita o uso, mas não é prático de ser acessado, podendo se tornar o procedimento oneroso.

O plano de qualidade define medições, estabelecidas na seção 5 do SQAP elas são resultados de SQA. O projeto em questão, gerador de uma entrega do tipo compilação, permitia medição a partir de defeitos e de erros. Utilizando o registro de tempo, volume de erros e classificação destes como dados a serem medidos, para avaliação do processo de SQA e do processo de desenvolvimento do Promob.

A seção 7 tratava do registro de não conformidades resultantes das atividades de testes e das tarefas de SQA foram realizadas no Portal de Solicitações, conforme proposta e eliminando o uso de planilhas.

Conforme seção 8 o registro de não conformidades foi realizado, no Portal de Solicitações da Promob.

Das ferramentas de apoio, listadas na seção 9, o MS Office Excel deixa de ser utilizado para elaboração dos *checklists*, sendo substituído pelo GoogleDocs, mas permanece na lista para gerar os relatórios, que também sofreram alteração, não sendo gerados automaticamente como previsto, devido a renovação da equipe responsável pela solução MS Sharepoint. A seção 10, Controle de mídia, pertence a área de atuação da equipe de TI. Não foi necessária a intervenção dela no projeto em questão.

A seção 11, que trata de registros, manutenção e retenção de documentos de SQA, foi parcialmente executada como planejado. As solicitações são criadas e ficam armazenados no servidor do Sharepoint, os demais documentos estão armazenados no servidor da U.D. Relatórios, que inicialmente seriam armazenados junto ao Sharepoint, não foram automaticamente gerados e, portanto também estão armazenados no servidor da U.D.

O momento de treinamento, seção 12, foi utilizado também para sensibilizar as pessoas. Foi a primeira atividade do projeto da compilação 4. Não houve um treinamento formal, adotando a mesma forma de disseminar informação aplicada na U.D. Primeiramente foi apresentado, via e-mail, as alterações que ocorreriam para o projeto de compilação que se inicializaria e depois, apresentado os *checklists*, bem como os porquês e objetivos dos mesmos, para cada pessoa envolvida, de todas as áreas. Treinamento foi executado conforme proposto.

A gestão de riscos, estabelecida na seção 13, foi modificada após a execução do projeto, adicionando o risco: Não preenchimento dos *checklist* de prioridade e gravidade alta, e como contingência deve ter a intervenção do superior imediato.

No glossário, seção 14, foi adicionado o termo Google Docs. E a seção 15 teve o quadro preenchido pela primeira vez, registrando que o mesmo sofreu mudanças para adequar as seções com as melhorias e lições aprendidas após finalizado o projeto de compilação, ao qual o SQAP foi aplicado.

5.3 APLICAÇÃO DAS MÉTRICAS

Julgar qualidade significa em essência interpretar os resultados das medições e obter conclusões sobre qualidade. (Koscianski, et. al., 1999). A aplicação das métricas revelou o resultado obtido na aplicação do SQPA em números.

- a) O primeiro resultado das medições a ser apresentado é a severidade dos defeitos. Na tabela 14 são apresentados as 53 solicitações reportadas no período em que a compilação 4 ficou em produção, classificadas pela severidade. A classificação é realizada pelo desenvolvedor que implementou a solução, e não é um campo obrigatório no preenchimento da solicitação e portanto, algumas solicitações não foram avaliadas.

TABELA 14 - Severidades dos Defeitos da Compilação 4

Severidade	Número de solicitações
Urgente	2
Alta	7
Média	16
Baixa	3
Melhoria	5
Não avaliada	20

b) Defeitos de Baixo impacto

Defeito de impacto baixo deveria ser obtido através da quantidade de solicitações de defeito, excluindo apenas as solicitações de melhorias, e que fossem priorizadas como “Média” e “Baixa”. A soma destas solicitações, dividida pelo número total de defeitos fornece o número de solicitações do impacto medido. O valor obtido para esta métrica, na compilação 4, foi de 0,35 solicitações de impacto médio.

c) Defeitos de médio impacto

Defeito de impacto médio deveria ser obtido através da quantidade de solicitações de defeito, excluindo apenas as solicitações de melhorias, e que fossem priorizadas como “Alta”. A soma destas solicitações, dividida pelo número total de defeitos fornece o número de solicitações do impacto medido. O valor obtido para esta métrica, na compilação 4, foi de 0,13 solicitações de impacto médio.

d) Defeitos de Alto impacto

Defeito de impacto alto deveria ser obtido através da quantidade de solicitações de defeito, excluindo apenas as solicitações de melhorias, e que fossem priorizadas como “Urgente”. A soma destas solicitações, dividida pelo número total de defeitos fornece o número de solicitações do impacto medido. O valor obtido para esta métrica, na compilação 4, foi de 0,04 solicitações de impacto alto.

e) Distribuição dos Erros por etapa

A métrica de distribuição dos erros (tabela 15) por etapa usa do preenchimento dos *checklists* para extrair seus dados. Um formulário preenchido mais de uma vez para um mesmo identificador é interpretado como um erro. Entretanto, nenhum formulário foi repetido para um mesmo identificador de requisito. Todos os erros registrados tem origem na execução de testes, 19 solicitações cadastradas, sendo uma melhoria. Portanto, 18 erros foram reportados.

TABELA 15 - Tabela da Distribuição dos Erros por etapa da compilação 4

Severidade	Número de solicitações
Planejamento	0
Análise e desenvolvimento	0
Documentação	0
Testes	18
Disponibilização	0

f) Severidade dos erros

Resultante da etapa de testes 19 solicitações foram reportadas. A tabela 16 apresenta a segmentação destas por severidade. A classificação é realizada pelo testador e é um campo obrigatório no preenchimento da solicitação.

TABELA 16 - Tabela da Severidade dos Erros da compilação 4

Severidade	Número de solicitações
Erro de funcionalidade	17
Erro de Padrão	0
Erro de Usabilidade	1
Erro TU	0

Severidade	Número de solicitações
Melhoria	1

g) Retrabalho no ciclo de desenvolvimento

A medição do retrabalho, durante a etapa de desenvolvimento, permite identificar o esforço que está sendo necessário devido aos erros não detectados nas etapas iniciais. A compilação 4 foi desenvolvida em 20 dias úteis, correspondente a um mês de trabalho da equipe de compilação ou 338 horas trabalhadas em melhorias e 702 horas em correção, totalizando 1040 horas. 101 horas foram necessárias para atender as solicitações geradas na execução de testes. A tabela 17 representa a distribuição de horas por projeto.

TABELA 17 - Horas de trabalho

Tipo de solicitação/requisito	Total de horas trabalhadas
Requisito de melhorias	338
Requisitos de correção	601
Solicitação de testes	101

5.4 ANÁLISE DE RESULTADOS

As informações fornecidas pela auditoria, juntamente com os números do projeto e com as métricas possibilitam realizar uma análise crítica, sobre os resultados obtidos com a aplicação do SQAP, para avaliar o processo de SQA e o processo de desenvolvimento do Promob.

Os resultados da compilação 4 serão comparados com a média histórica dos três projetos de mesmo tipo, compilação, imediatamente anteriores ao projeto da compilação em questão. A escolha de três projetos para compor a média dá-se porque estes foram monitorados e, portanto possui dados confiáveis.

O projeto da compilação 4 foi composto por 67 requisitos, quando 66 é a média, portanto 1 requisito a mais que a média (tabela 18).

A compilação 4 foi desenvolvida em 20 dias úteis. A média de tempo para o desenvolvimento de um projeto de compilação é 27, 7 dias (tabela 19).

Na avaliação do tempo de implementação por requisito ou solicitação, que na média é 15,6 horas, conforme tabela 20. Para a compilação 4, que adicionou as atividades de SQA, utilizou-se 12,7 horas, valor abaixo da média mostra que o processo não foi onerado.

TABELA 18 - Tabela de requisitos

Requisitos	Compilação 1	Compilação 2	Compilação 3	Média	Compilação 4
Requisitos de erros	49	38	46	44	54
Requisitos de melhorias	25	16	21	21	13
Total	74	54	69	66	67

TABELA 19 - Tempo de desenvolvimento de compilações

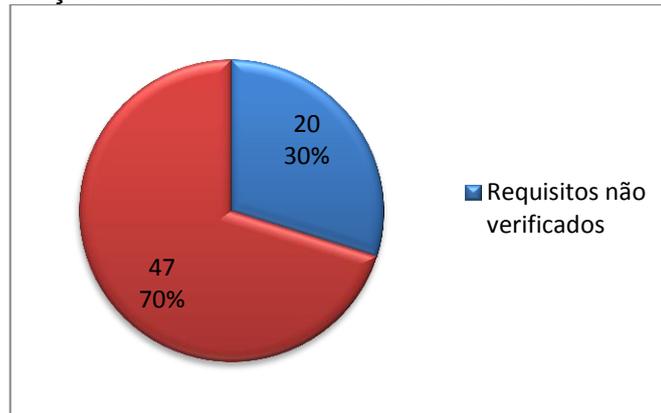
Compilação	Dias trabalhados
Compilação 1	26
Compilação 2	30
Compilação 3	23
Média	26,3
Compilação 4 (piloto)	20

TABELA 20 - Tempo médio por solicitação

Compilação	Dias trabalhados	Horas trabalhadas (total)	Nº requisitos atendidos (total)	Tempo médio por solicitação
Compilação 1	26	1352	111	12,2
Compilação 2	30	1560	79	19,7
Compilação 3	23	1196	81	14,8
Média	26,3	1369	90,3	15,6
Compilação 4	20	1040	82	12,7

A métrica distribuição dos erros por etapa revela a participação do SQAP no processo de desenvolvimento. Porém, todos os erros registrados tem origem na execução de testes que cadastrou 19 solicitações, sendo uma delas uma solicitação de severidade melhoria. Portanto, 18 erros foram reportados. Entretanto, informações relevantes foram extraídas a partir da auditoria realizada.

A auditoria não identificou inconformidades na etapa de planejamento, mas identificou 20 não conformidades na etapa de análise e implementação. Vinte requisitos, do total de 67 atendidos na compilação, não foram verificados, ou seja, 30% dos deles não foram verificados (figura 33).

FIGURA 33 – Execução do *checklist* de análise e do *checklist* de implementação

Fonte: próprio autor

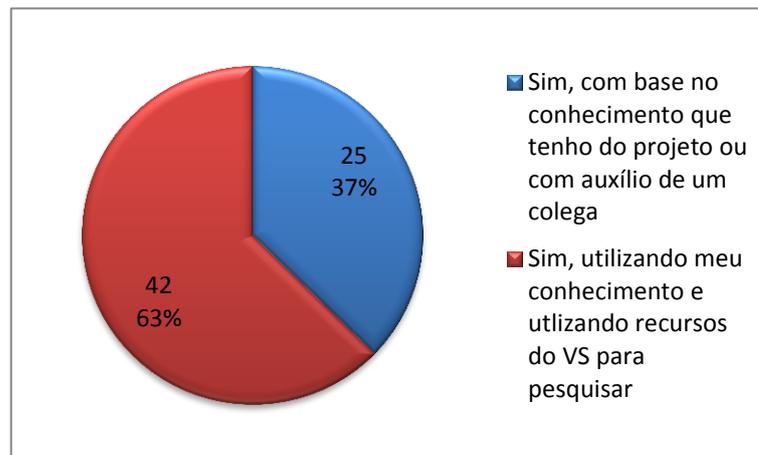
Sobre os motivos da não execução da verificação, os desenvolvedores informaram que o não preenchimento foi por motivo de esquecimento. A tabela 21, mostra que 7 (sete) dos 8 (oito) desenvolvedores responsáveis pela compilação esqueceram de avaliar alguma solicitação e todos avaliaram pelo menos uma.

TABELA 21 - Requisitos verificados

Desenvolvedores	Solicitações implementadas	Verificações realizadas	Verificações não realizadas
Desenvolvedor 1	7	5	2
Desenvolvedor 2	4	3	1
Desenvolvedor 3	8	2	6
Desenvolvedor 4	25	24	1
Desenvolvedor 5	9	1	8
Desenvolvedor 6	9	8	1
Desenvolvedor 7	2	1	1
Desenvolvedor 8	3	3	0

Dos 70% dos *checklist* de análise (anexo C) e *checklist* de implementação (anexo D) preenchidos é possível extrair informações como a apresentada na figura 30 que mostra o resultado da questão: “3.analise: Os relacionamentos e dependências com a alteração a ser realizada foram avaliados?”, cujas respostas mostram que 37% das solicitações são atendidas com consulta a um colega (figura 34).

FIGURA 34 – Conhecimento para análise



Fonte: próprio autor

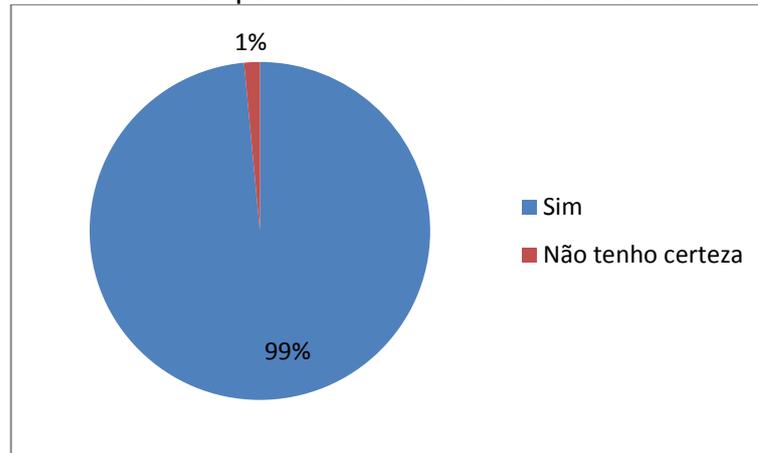
Para esta mesma questão, analisando a experiência do desenvolvedor através do tempo de contratação deste pela empresa. A tabela 22 mostra que este critério não é um fator determinante para a independência na solução das solicitações/requisitos.

TABELA 22 - Experiência do desenvolvedor

Os relacionamentos e dependências com a alteração a ser realizada foram avaliados?	Mais de 3 anos	Menos de 3 anos
Sim, com base no conhecimento que tenho do projeto ou com auxílio de um colega	19	6
Sim, utilizando meu conhecimento e utilizando recursos do VS para pesquisar	33	9

Outra questão diz: “2.analise: Você conhece o contexto e/ou a aplicação do requisito que está alterando, sendo capaz de formular um caso de teste?” mostra um resultado (figura 31) bastante satisfatório. Apenas para uma solicitação o desenvolvedor respondeu que Não tinha certeza se era capaz (figura 35).

FIGURA 35 – Capacidade de criar caso de teste

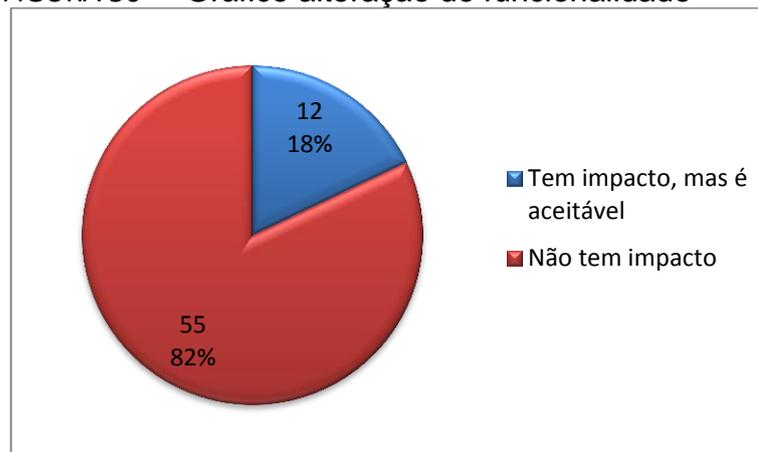


Fonte: próprio autor

Através da questão: “4.analise: O impacto da alteração a ser realizada é aceitável? Não altera funcionamento/comportamento da(s) funcionalidades existentes?” é possível identificar que 18% das solicitações atendidas tem impacto nas funcionalidades dos produtos (figura 36).

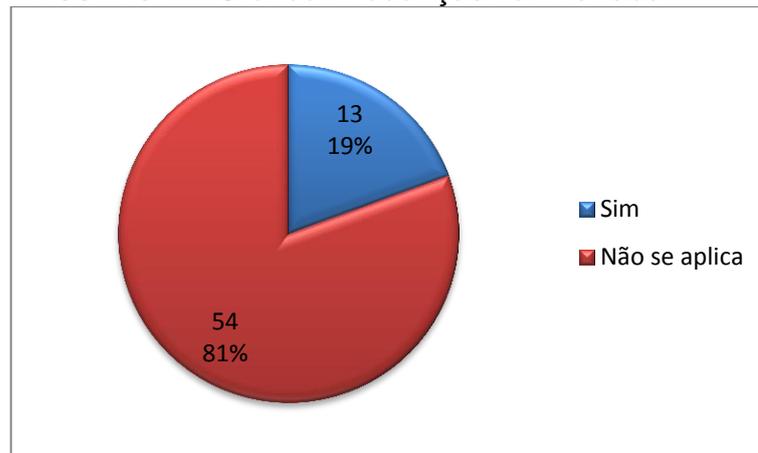
A questão de implementação: “1.interface: A interface proposta segue os padrões acordados para o produto?” possibilita avaliar as alterações de interface. Conforme figura 37, 19% das solicitações necessitam de alteração de interface, a relevância deste percentual está que usabilidade é a característica mais desejada para o produto Promob.

FIGURA 36 – Gráfico alteração de funcionalidade



Fonte: próprio autor

FIGURA 37 – Gráfico mudanças na interface

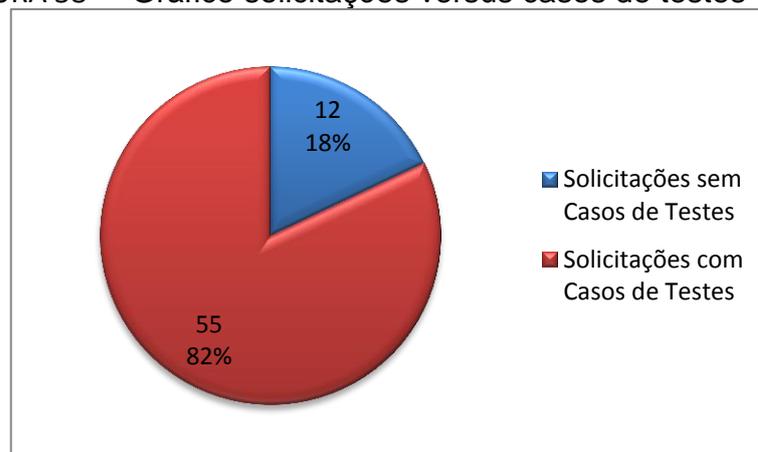


Fonte: próprio autor

A etapa de testes gera os dados para a métrica Severidade dos Erros, mas é possível analisá-la desde o planejamento com a criação dos casos de testes. No documento de Casos de testes 12 (doze) não conformidades foram identificadas, ou seja, a auditoria revelou que doze requisitos não tiveram casos de testes planejados, embora estivessem os requisitos listados no documento. O que corresponde a 18% (dezoito) dos requisitos do projeto sem planejamento de teste, conforme figura 38.

Os motivos para o não planejamento, verificado junto aos responsáveis por esta tarefa, identificou que quando a solicitação é muito específica, o caso de teste seria repetição da descrição da solicitação, logo para ganhar tempo o mesmo não é descrito, sendo seguida a descrição da solicitação.

FIGURA 38 – Gráfico solicitações versus casos de testes

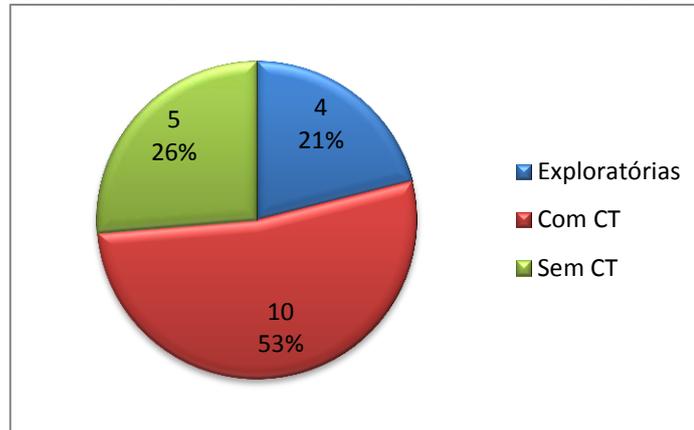


Fonte: próprio autor

O total de solicitações resultantes da execução de testes é apresentado na figura 39. São 19 (dezenove) solicitações, segmentadas em solicitações geradas a

partir de requisitos para os quais casos de testes foram criados, para os quais não foram criados e para solicitações exploratórias. Considera-se exploratória aquele erro que ocorre devido alterações do projeto em execução, porém não tem associação direta ao requisito atendido.

FIGURA 39 – Gráfico revisão de casos de testes



Fonte: próprio autor

A média de solicitações geradas na execução de testes é de 27 solicitações (tabela 23), ou seja, em média são geradas 0,4 solicitações por requisito. Neste projeto, o número de solicitações de testes foi de 19, que corresponde a 0,28 solicitações por requisito. Reduzindo em 30% o número de solicitações de testes.

TABELA 23 - Solicitações de Testes

Solicitações de testes	Compilação 1	Compilação 2	Compilação 3	Média	Compilação 4
Erro de funcionalidade	26	9	7	14	17
Erro de Padrão	3	8	3	5	0
Erro de Usabilidade	2	2	2	2	0
Erro TU	10	5	0	5	1
Melhoria	0	4	0	1	1
Total	41	28	12	27	19

Os erros identificados em testes são apresentados na métrica Severidade dos Erros, classificados em quatro categorias, que auxiliam a equipe de desenvolvimento a estabelecer prioridades e o tipo TU – Teste Unitário é monitorado pelo gerente da U.D. Na compilação em questão, houve apenas 1 ocorrência de erro classificado como TU, quando a média é 5 solicitações. Um também foi o número de solicitações de usabilidade e de melhorias. A categoria de erro de padrão não teve ocorrência de

erro identificada. Já o número de erros nas funcionalidades foi 17, quando a média é 14 solicitações.

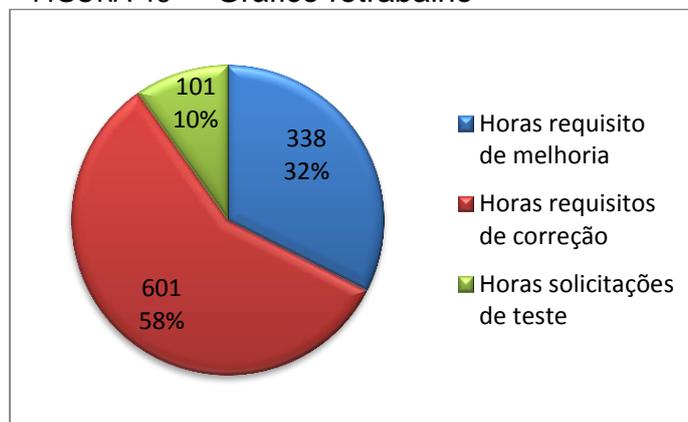
O aumento de erros de funcionalidade confirma uma situação bastante discutida na empresa, desde que a mesma adotou as práticas de testes de *software*, que quanto menos erros simples forem identificados, mais erros de severidade alta poderiam ser identificados, pois a equipe de testes poderia melhor utilizar o seu tempo.

A métrica, Retrabalho no Ciclo de Desenvolvimento, permite identificar o esforço que está sendo necessário devido aos erros não detectados nas etapas iniciais. Na figura 40 é possível observar que 10% (dez) do tempo de desenvolvimento da compilação é direcionado a correção de erros reportados após a execução de testes.

Finalizado o projeto, a compilação entra em produção. Não incomum, um erro de alta severidade é reportado pela equipe de atendimento ao cliente ou pela equipe de personalização. A correção deste dispara uma nova entrega, da mesma compilação. Este novo ciclo da compilação, provoca alterações no cronograma de trabalho. A equipe de desenvolvimento, testes e documentação interrompe suas atividades planejadas para realizar a nova entrega da compilação. Este é um processo crítico, pois fazer com que a correção chegue ao cliente final é oneroso e demorado.

Alterações na forma de distribuição da compilação estão sendo providenciadas para que deixe de ser oneroso e demorado. Entretanto, novas liberações de mesma compilação é sempre algo a ser evitado e somente realizado em casos de extrema necessidade.

FIGURA 40 – Gráfico retrabalho



Fonte: próprio autor

A tabela 24 apresenta a média do número de publicações realizados para as últimas entregas e para a compilação para o qual o SQAP foi aplicado. O resultado foi de 2 publicações, quando a média é de 4. Logo, redução de 50%.

TABELA 24 - Número de Publicações

Entregas	Compilação 1	Compilação 2	Compilação 3	Média histórico	Compilação 4
Publicações	4	2	6	4	2

A métrica, Priorização de Defeitos, identificou que a compilação 4 gerou 53 solicitações de defeito, 7,5% a mais que a média, conforme tabela 25. Observa-se na tabela o total de defeitos gerados e o número de solicitações geradas pela CA – Central de Atendimento – que atende o usuário final, onde o impacto de qualquer defeito é maior, pois atinge um grande número de clientes, e pode necessitar da disponibilização da compilação com a correção. Este número foi reduzido de 20,7 para 18. Embora o número de solicitações tenha sido maior que a média, um número menor de erros foi percebido pelo cliente.

TABELA 25 - Número de defeitos

Projeto	Nº solicitações cadastradas (defeito)	Nº solicitações cadastradas (defeito) - Só Suporte
Compilação 1	73	29
Compilação 2	21	11
Compilação 3	54	22
Média	49,3	20,7
Compilação 4	53	18

Pela análise de impacto, das métricas Defeitos de Baixo impacto, Defeitos de Médio impacto e Defeitos de Alto Impacto, excluindo as solicitações não avaliadas, observa-se os índices da tabela 26. Para defeitos de impacto alto, que são aqueles que impedem o usuário de trabalhar ou de utilizar uma funcionalidade vital a redução foi de 54 %. Os defeitos de baixo impacto reduziram 16 % e aumentou outros 16% no número de solicitações de impacto médio. Este resultado reflete o resultado das solicitações de testes, erros simples como padrão não foram identificados e erros graves reduziram.

TABELA 26 - Impacto dos defeitos

Projeto	Baixo impacto	Médio impacto	Alto impacto
Compilação 3	0,68	0,18	0,13
Compilação 4	0,57	0,21	0,06

5.3 CONSIDERAÇÕES DA APLICAÇÃO

Resultado do planejamento do processo de garantia, o Plano de Garantia da Qualidade foi estabelecido, definindo estratégia, técnicas, métricas e artefatos para atuação, conforme práticas descritas nos modelos CMMI, MPS.BR e ISO/IEC 12207. O SQAP proposto e aceito pelo gerente da U.D. Projetos foi aplicado a um projeto real.

A aplicação do Plano possibilitou realizar o processo de Garantia da Qualidade no processo de desenvolvimento de *software* Promob. Todas as atividades propostas, que cobriam todas as etapas do ciclo de desenvolvimento, foram realizadas, embora nem todas com sucesso.

A proposta previa realizar medições, estabelecidas na seção 5 do SPQA, para qual a extração dos dados deveria ser automatizada. O que não ocorreu, por falta de disponibilidade dos responsáveis para realização da tarefa visto que houve renovação do quadro de colaboradores da equipe. Extrair os dados sem automatização se tornou bastante oneroso.

As medições permitiram identificar que o total de solicitações ficou bem abaixo da média, 19 solicitações de testes, identificadas na compilação 4, contra 27 solicitações na média. Já o número de defeitos ficou acima da média, que é 49,3 solicitações, e 53 foram reportadas durante o período de produção da compilação. Porém, o número de defeitos reportado pela Central de Atendimento decresceu, então menos erros, ou erros de menor severidade, foram percebidos pelo cliente final. Ao avaliar a severidade das solicitações há confirmação do reflexo dado pelo número de solicitações reportadas pela Central, identificou-se a redução de mais de 50% das solicitações de impacto alto e, a redução do número de solicitações de impacto baixo. Este último refletindo os mesmos resultados identificados nas solicitações dos testes, a redução de erros/falhas simples, como erros de padrão, usabilidade e falta de teste unitário.

As métricas do plano forneceram pela primeira vez dados resultantes do processo de desenvolvimento para a empresa. Os dados obtidos e as informações construídas a partir das medições fornecem uma base sólida para avaliações e análises críticas. Ao mesmo momento que possibilita identificar pontos de melhoria no processo, a fim de agregar qualidade ao Promob e resolver o problema que deu origem a este estudo.

6. CONCLUSÃO

Trabalhar qualidade depende de muitas variáveis, as imprescindíveis são: um ambiente favorável, no qual todos os envolvidos devem conhecer e estar cientes de que não se trata de um procedimento de detecção de erros e sim da prevenção deles, o entendimento da importância de ter qualidade em todos seus processos e produtos; e o apoio da alta gerencia, para realizar investimentos necessários e possibilitar que a avaliação de qualidade seja realizada com total imparcialidade. Além de desejável, o apoio pode ser determinante para os resultados obtidos. Este projeto foi apoiado, mas não acompanhado, pelo cargo de maior hierarquia na U.D. Projetos, o gerente da mesma. A adesão da equipe foi satisfatória, considerando que houve mais de 70% de participação na tarefa mais crítica, a revisão de análise e implementação. E o projeto pôde ser avaliado com imparcialidade.

O plano foi proposto conforme modelo da norma IEEE 730, o modelo é estrutural, porém as seções foram tornando necessárias as definições das demais normas estudadas. Por exemplo, a seção 1, deveria estabelecer as características de qualidade para o *software*. Estas foram identificadas através de pesquisa com a equipe Promob, através das definições da ISO/IEC 9126-1, e também utilizando a Pesquisa de Satisfação, realizada por com clientes. A seção 4, que tratava de documentação, para os quais os artefatos foram elaborados, como o plano de V&V, e que tem como referência os resultados esperados do modelo MPS.BR e o modelo CMMI. Seção 5 que previa as medições, para a qual utilizou-se a ISO/IEC 9126-2 para estabelecer as métricas de produto e as orientações de Bartié (2002), Pressman (2006) e Sommerville (2007). E assim sucessivamente para as demais seções.

Através da proposta resultante deste estudo, o conceito de qualidade foi aplicado. Revisões dos produtos de *software* das fases intermediárias foram determinadas. Aplicado a um projeto real, o plano foi seguido, produtos revisados, em todas as etapas do processo de desenvolvimento, através dos *checklists*. Resultando no registro e comunicação de não conformidades, detectadas durante as atividades de SQA, que foram medidas e avaliadas.

As atividades executadas por apenas uma pessoa, como as atividades de documentação, tem chances menores de erros ou de serem negligenciadas. Quanto maior a equipe, maior o risco de tarefas não serem executadas. Motivo pelo qual as

tarefas planejadas não foram executadas para todos os requisitos. Quando não executadas o motivo apresentado foi esquecimento, com a justificativa de que como era uma nova atividade e não estavam habituados. Há que se ressaltar que é um ponto forte o comprometimento da equipe com todos os projetos e existe boa relação entre as equipes de desenvolvimento e testes, áreas onde comumente existem conflitos. Sendo o SQAP de responsabilidade da sub unidade Núcleo de Testes a relação já construída favoreceu a colaboração dos envolvidos com o projeto.

O objetivo central do processo de SQA é evitar a disseminação dos erros, de uma etapa para as seguintes, durante o ciclo de desenvolvimento do produto. Este não foi diretamente comprovado, visto que erros foram registrados apenas na etapa de testes. Entretanto, houve uma redução no número de solicitações resultantes da etapa de testes de 0,40 solicitações por requisito para 0,28, ou seja, uma redução de 30%. E um depoimento, ao finalizar o projeto piloto, se destaca. Um dos desenvolvedores revela: “sempre que finalizava um requisito, sabendo que deveria responder o *checklist*, tentava recordar as questões e questionava se havia realizado o tratamento de exceções, se tinha certeza que conhecia a necessidade etc.”. O depoimento permite extrair duas informações: realizar o processo de revisão já estava sendo uma atividade rotineira, e preocupar-se com a qualidade do produto pelo qual era responsável também passava a ser mais natural.

O principal objetivo deste trabalho era a elaboração de um Plano de Garantia da Qualidade para resolver o problema diagnosticado no ciclo de desenvolvimento da U.D. Projetos, para o produto Promob, a sobrecarga as equipes ao chegar o produto na etapa de testes, onde eram identificados todos os erros. O SQAP tinha como premissa não alterar o processo de desenvolvimento existente, e assim foi proposto e executado.

Como contribuição, o trabalho forneceu as características de qualidade desejadas, documentos como o desenho atualizado do processo de desenvolvimento da U.D, o Plano de SQA, artefatos como Plano de V&V, métricas para avaliar o produto, o processo de desenvolvimento e o processo de SQA e um case do processo de SQA, que foi aplicado a uma das quatro unidades de desenvolvimento da empresa.

Para futuros trabalhos, as medições necessitam ser automatizadas, desonerando a obtenção dos dados para as métricas. Sobre estas a construção de

indicadores poderia ser estabelecidas. No processo, uma forma de alertar os responsáveis sobre as revisões, preenchimento dos *checklists*, eliminaria o problema identificado no projeto piloto, o esquecimento de realizar a revisão. A revisão do processo de desenvolvimento fica como recomendação, com alta criticidade para a documentação dentro de cada atividade do processo, atualmente praticamente todo o conhecimento existente na U.D. Projetos é tácito.

REFERÊNCIAS

ABNT – ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ISO/IEC 12207 – Tecnologia da Informação – Processos de ciclo de vida de *software***. Rio de Janeiro: ABNT, 1998.

ABNT – ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126 – Engenharia de software - Qualidade de produto**. Rio de Janeiro: ABNT, 2002.

ABNT – ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ISO/IEC 14598 – Engenharia de software – Avaliação de produto**. Rio de Janeiro: ABNT, 2001.

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. **MPS.BR – Guia de Implementação Parte 2**, junho 2011. Disponível em: <<http://www.softex.br>>. Acessado em: 15 de maio, 2011.

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. **MPS.BR – Guia de Implementação Parte 4**, junho 2011. Disponível em: <<http://www.softex.br>>. Acessado em: 15 de maio, 2011.

BARTIÉ, ALEXANDRE. **Garantia de Qualidade de *Software***. 5.ed. Rio de Janeiro, RJ, Brasil. Elsevier, 2002.

IEEE Std 1028-1997. **IEEE Standard for *Software Reviews***: IEEE, 1997.

IEEE Std 1061-1998. **Standard for a *Software Quality Metrics Methodology***: IEEE, 1998.

IEEE Std 730 – 1998. **Standard for *Software Quality Assurance Plans***. IEEE, 2008

KATSURAYAMA, ANNE ELISE. **Apoio à Garantia da Qualidade do Processo e do Produto em Ambientes de Desenvolvimento de *Software Orientados a Organização***. Rio de Janeiro: UFRJ, 2008. 160 p. Dissertação (Mestrado) - Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, 2008.

KOSCIANSKI, A.; VILLAS-BOAS, A.; REGO, C. M.; ASANOME, C.; SCALET, D.; ROMERO, D.; CIESLAK, J. M.; PALUDO, M.; FROSSARD, R. S.; VOSTOUPAL, T.M. **Guia para Utilização das Normas Sobre Avaliação de Qualidade de Produto de *Software* – ISO/IEC 9126 e ISO/IEC 14598**. Curitiba, maio 1999.

LAZZARI, **Proposta de um Processo de Garantia Da Qualidade conforme Modelo MPS.BR Nível F**. Centro de Computação e Tecnologia da Informação, Universidade de Caxias do Sul: Caxias do Sul: 2009. Bacharelado em Ciência da Computação – Trabalho de Conclusão de Curso.

MATARAZZO, DANTE CARMINE. **Análise financeira de balanços: abordagem básica e gerencial**. 6. ed. São Paulo: Atlas, 2003.

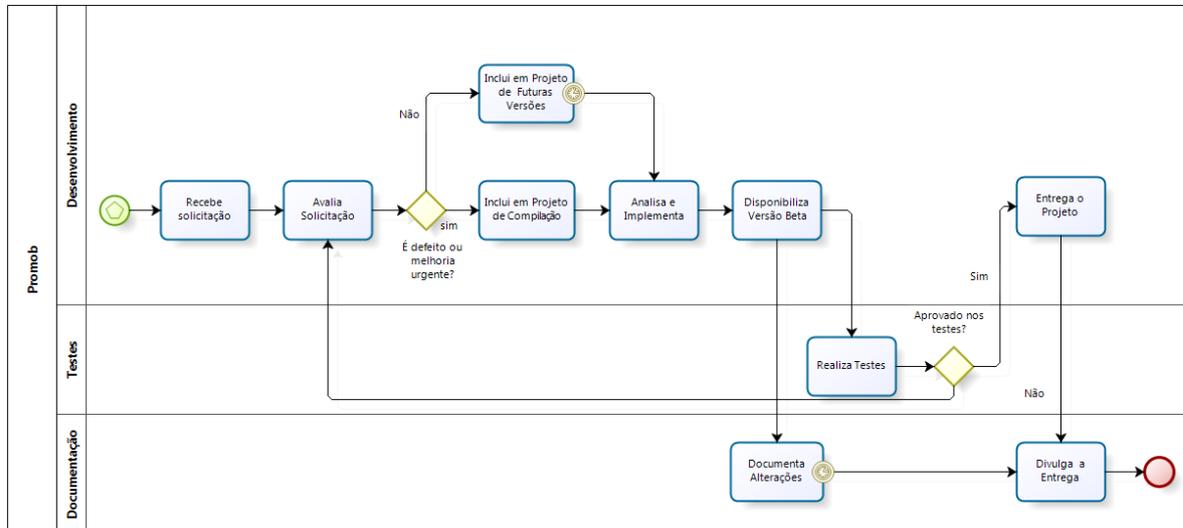
PRESSMAN, Roger S. **Engenharia de Software**. McGraw-Hill, São Paulo, 2006.

ROCHA, A.R. J.C. MALDONADO; K.C. WEBER, (2001) **Qualidade de Software – Teoria e Prática**. Prentice Hall, São Paulo.

SOMMERVILLE, I. (2007) **Engenharia de Software**. Addison Wesley, 8ª Edição.

ANEXOS

ANEXO A – Processo de desenvolvimento do Promob



- **Recebe Solicitação:** a solicitação cadastrada no Portal de Solicitações é recebida pelo responsável por atendê-la e/ou providenciar seu atendimento
- **Avalia Solicitação:** a solicitação é avaliada, podendo ser encaminhada para desenvolvimento ou colocada em espera e incluída em futuras versões do produto.
- **Inclui em Projeto de Futuras Versões:** a solicitação avaliada, classificada como uma melhoria e colocada em espera para ser implementada em futuras versões do produto.
- **Inclui em Projeto de Compilação:** a solicitação avaliada, reporta um erro ou uma nova necessidade de um cliente, que não pode aguardar até uma nova versão, esta solicitação entra no projeto de compilação, que possui entregas mensais.
- **Analisa e Implementa:** a solicitação é analisada e implementada pelo desenvolvedor a qual foi atribuída. As informações que o desenvolvedor necessita para executar a tarefa são repassadas de forma verbal e informal pelo gerente da U.D.;
- **Disponibiliza versão beta:** desenvolvedor responsável pela gerencia de configuração entrega um pacote com todas as solicitações que já foram atendidas no período.

- Documentar alterações: equipe de documentação cria material a ser disponibilizado para usuários internos e externos sobre a solução implementada;
- Realizar testes: Núcleo de Testes cria casos de testes e os executa para validar as implementações realizadas, cadastrando solicitações para as não aprovadas.
- Entrega o projeto: o produto é disponibilizado no diretório pré-definido e os projetos .project e o da compilação fechados, respectivamente no Project Server e no Forrest.
- Divulga a entrega: Por último, ocorre a divulgação da entrega do produto de versão/compilação comunicado através do blog da U.D. Projetos.

ANEXO B – Pesquisa de Qualidade

1 - Conforme a importância/relevância das subcaracterísticas de Funcionalidade classifique-as numerando de 1 a 5 *

Atribua 1 para a característica mais relevante e 5 e menos relevante. Não repita o valor atribuído para diferentes características.

	1	2	3	4	5
Adequação: presença das funções especificadas	<input type="radio"/>				
Acurácia: o produto gera resultados precisos ou dentro do esperado	<input type="radio"/>				
Interoperabilidade: capacidade de interagir e interoperar com outros sistemas, de acordo com o especificado.	<input type="radio"/>				
Conformidade: observância a padrões, convenções ou regras estabelecidas.	<input type="radio"/>				
Segurança de acesso: capacidade para prevenir o acesso não autorizado.	<input type="radio"/>				

2 - Conforme a importância/relevância das subcaracterísticas de Confiabilidade classifique-as numerando de 1 a 3. *

Atribua 1 para a característica mais relevante e 3 e menos relevante. Não repita o valor atribuído para diferentes características.

	1	2	3
Maturidade: indicação de baixa frequência de falhas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tolerância a falhas: capacidade do produto para manter determinados níveis de desempenho mesmo na presença de problemas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Recuperabilidade: capacidade do produto para reestabelecer o nível de desempenho desejado e recuperar dados em caso de ocorrência de falha	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. Conforme a importância/relevância das subcaracterísticas de Usabilidade classifique-as numerando de 1 a 3. *

Atribua 1 para a característica mais relevante e 3 e menos relevante. Não repita o valor atribuído para diferentes características.

	1	2	3
Inteligibilidade: medida da facilidade do usuário para reconhecer a lógica de funcionamento do produto e sua aplicação.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apreensibilidade: medida da facilidade encontrada pelo usuário para aprender a utilizar o produto.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Operacionalidade: medida da facilidade para operar o produto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. Conforme a importância/relevância das subcaracterísticas de Eficiência classifique-as numerando de 1 a 2. *

Atribua 1 para a característica mais relevante e 2 e menos relevante. Não repita o valor atribuído para diferentes características.

	1	2
Comportamento com relação ao tempo: medida do tempo de resposta e de processamento, assim como as taxas de processamento, ao executar as funções prescritas.	<input type="radio"/>	<input type="radio"/>
Comportamento com relação ao uso de recursos: medida da quantidade de recursos necessários (CPU, disco e memória etc).	<input type="radio"/>	<input type="radio"/>

5. Conforme a importância/relevância das subcaracterísticas de Manutenibilidade classifique-as numerando de 1 a 4. *
Atribua 1 para a característica mais relevante e 4 e menos relevante. Não repita o valor atribuído para diferentes características.

	1	2	3	4
Analisabilidade: medida do esforço necessário para diagnosticar deficiências ou causas de falhas, ou localizar as partes a serem modificadas para corrigir os problemas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Modificabilidade: medida do esforço necessário para realizar alterações, remover falhas ou para adequar o produto a eventuais mudanças de ambiente operacional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Estabilidade: medida do risco de efeitos inesperados provenientes de modificações.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testabilidade: medida do esforço necessário para testar o software alterado.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. Conforme a importância/relevância das subcaracterísticas de Portabilidade classifique-as numerando de 1 a 4. *
Atribua 1 para a característica mais relevante e 4 e menos relevante. Não repita o valor atribuído para diferentes características.

	1	2	3	4
Adaptabilidade: ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Capacidade para ser instalado: ser instalado em um ambiente especificado.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Coexistência: coexistir com outros produtos de software independentes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Capacidade para substituir: ser usado em substituição a outro produto de software especificado.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7 - Numere as características abaixo, de forma a classificar a característica mais e menos importante/relevante para o Promob. *

Atribua 1 para a característica mais relevante e 5 e menos relevante. Não repita o valor atribuído para diferentes características.

	1	2	3	4	5
Funcionalidade (Satisfação das Necessidades)	<input type="radio"/>				
Confiabilidade (Imunidade a Falhas)	<input type="radio"/>				
Usabilidade (Facilidade de Uso)	<input type="radio"/>				
Eficiência (Rápido e "Enxuto")	<input type="radio"/>				
Manutenibilidade (Facilidade de Manutenção)	<input type="radio"/>				

8. Selecione a posição (cargo) que ocupa na empresa *

Outro cargo

ANEXO C – Checklist de análise**Análise**

Informe o ID da solicitação para a qual está respondendo este formulário.*

1.analise: A solução proposta, além de solucionar o problema apresentado, atende a necessidade do cliente?*

- Não conheço a necessidade do cliente
- Sim

2.analise: Você conhece o contexto e/ou a aplicação do requisito que está alterando, sendo capaz de formular um caso de teste?*

- Não
- Não tenho certeza
- Sim

3.analise: Os relacionamentos e dependências com a alteração a ser realizada foram avaliados?*

- Não
- Sim, com base no conhecimento que tenho do projeto ou com auxílio de um colega
- Sim, utilizando meu conhecimento e utilizando recursos do VS para pesquisar

4.analise: O impacto da alteração a ser realizada é aceitável? Não altera funcionamento/comportamento da(s) funcionalidades existentes?*

- Não tem impacto
- Tem impacto, mas é aceitável
- Desconheço o impacto que pode ter

Enviar

ANEXO D – Checklist de implementação

Código fonte

Informe o ID da solicitação para a qual está respondendo este formulário.*

1.implem: O teste unitário foi realizado? A solução implementada funciona corretamente?*

- Sim
 Não

2.implem: Tratamentos de erros (!= null, exists....) foram tratados?*

- Sim
 Não
 Não se aplica

3.implem: O código produzido está organizado, legível e seguindo os padrões adotados pela UD?*

- Sim
 Não
 Sim, com base no conhecimento que tenho do projeto ou com auxílio de um colega
 Sim, utilizando meu conhecimento e utilizando recursos do VS para pesquisar

4.implem: A alteração foi realizada em todas as versões em desenvolvimento (current e compilações em aberto)?*

- Sim
 Não

5.implem: Os arquivos do projeto já foram sincronizados?*

- Sim
 Não

6.implem: A alteração realizada foi submetida (comitada)?*

- Sim
 Não

1.interface: A interface proposta segue os padrões acordados para o produto?

- Sim
 Não
 Não se aplica

2.interface: Nenhuma mensagem apresentada pelo sistema ao usuário final possui erros gramaticais?

- Sim
 Não
 Não se aplica

3.interface: Todas as mensagens e alteração de interfaces foram traduzidas para os idiomas espanhol e inglês?

- Sim
 Não
 Não se aplica

Enviar

ANEXO E – Checklist de requisito implementado**Requisito Implementado**

A documentação segue o template?

Sim

Não

O documento segue os padrões de formatação?

Sim

Não

O documento está no repositório pré-determinado?

Sim

Não

A linguagem do texto está adequada ao entendimento do cliente?

Sim

Não

Pergunta de exemplo 2

ANEXO F – Checklist de casos de testes

Casos de Testes

*Obrigatório

1. Os casos de testes estão claros e foram objetivamente definidos? *

- Sim
- Não

2. Os tipos de testes são focados nas características de qualidade desejáveis para o software? *

- Sim
- Não

3. Os casos de testes contemplam os tipos de testes priorizados para o projeto? *

- Sim
- Não

4. Os casos de testes possuem todos os cenários necessários para identificação de falhas? *

- Sim
- Não

Enviar

ANEXO G – Checklist do plano de produto

Plano de Produto

*Obrigatório

1. Todas as solicitações de alta prioridade no período forma atendidas? *

Sim

Não

2. Todas as solicitações atendidas para o projeto foram incluídas no plano de projeto? *

Sim

Não

3. Os recursos dos requisitos foram atribuídos no projeto? *

Sim

Não

4. As solicitações atendidas foram concluídas no portal de solicitações? *

Sim

Não

5. O plano de projeto atualizado foi publicado? *

Sim

Não

Enviar

ANEXO H – Checklist de Entrega**Disponibilização**

Todos os códigos-fontes alterados para o projeto foram submetidos?

Sim

Não

O executável foi gerado corretamente, com todos os arquivos necessários para a entrega?

Sim

Não

O código está ofuscado?

Sim

Não

O executável foi movido para o repositório pré-estabelecido?

Sim

Não

Os produtos dependentes foram gerados?

Sim

Não

O diretório das releases foram renomeados?

A descrição "Beta" deve ser removida.

Sim

Não

Pergunta de exemplo 2

Enviar

ANEXO I – Checklist de Auditoria

Auditoria do Processo de SQA

1. As revisões da etapa de planejamento foram executadas para todos os requisitos?

Aprovado

Reprovado

2. As revisões da etapa de análise e implementação foram executadas para todos os requisitos?

Aprovado

Reprovado

3. As revisões da etapa de documentação foram executadas para todos os requisitos?

Aprovado

Reprovado

4. As revisões da etapa de testes foram executadas para todos os requisitos?

Aprovado

Reprovado

5. As revisões da etapa de disponibilização foram executadas para todos os requisitos?

Aprovado

Reprovado

Enviar

ANEXO J – Plano de Garantia da Qualidade

Elaborado por:		Data:	
Revisado por:	Data:	Número de Revisão	Aprovado por:

Este documento apresenta o Plano de Garantia da Qualidade para o *software* Promob. Divido em 16 seções ele define os artefatos sob a Garantia da Qualidade e suas formas de avaliação.

Seção 1 – Propósitos

Esta seção apresenta os objetivos deste documento e o seu escopo.

1.1 Objetivos

O plano de qualidade do Promob tem como objetivos:

- Ser um guia para atingir os níveis de qualidade desejáveis;
- Estabelecer o consenso sobre os atributos de qualidade do produto Promob;
- Difundir os conceitos de garantia da qualidade de *software* para os envolvidos, conscientizando da importância da qualidade estar presente em todas as etapas de desenvolvimento;
- Sistematizar as atividades de SQA no processo de desenvolvimento do Promob.

O quadro a seguir prioriza as características e subcaracterísticas de qualidade desejadas para o *software* Promob.

Características de Qualidade	Subcaracterísticas de Qualidade	Prioridade da característica	Prioridade da subcaracterística
Usabilidade	Compreensibilidade	3	4
Usabilidade	Aprendizabilidade	3	3
Usabilidade	Operacionalidade	3	2
Usabilidade	Atratividade	3	1
Funcionalidade	Adequação	2	4
Funcionalidade	Segurança de acesso	2	3
Funcionalidade	Interoperabilidade	2	2
Funcionalidade	Acurácia	2	1
Eficiência	Comportamento de medidas de tempo	1	1
Eficiência	Medidas de utilização de recursos	1	2

1.2 Escopo

1.2.1 Do Produto

As recomendações referentes ao produto aplicam-se, exclusivamente, ao desenvolvimento do núcleo do *software* Promob, que contempla ferramentas e funcionalidades do produto.

O Promob é uma ferramenta de projeto, que possibilita reproduzir um layout de ambiente virtual. Contempla o projeto, o orçamento, o pedido e a documentação técnica do mesmo. Seu público são as lojas de móveis, arquitetos, marceneiros e designs de móveis.

Os procedimentos estabelecidos preveem o processo de desenvolvimento dos dois tipos de entregas de produto desenvolvido, realizadas pela U.D. Projetos:

- Versão: produto de *software* que possui novas ferramentas e funcionalidades e é disponibilizado anualmente;
- Compilação: produto de *software* que possui ajustes, correções e novas ferramentas e funcionalidades solicitadas por clientes em caráter de urgência. É entregue mensalmente;

1.2.2 Do Processo

O processo de desenvolvimento do *software* Promob será coberto pela garantia da qualidade nas etapas de:

- Planejamento: formar equipes de trabalho e selecionar requisitos/solicitações que serão atendidos no projeto;
- Análise e implementação: inclui atividades de planejamento e desenvolvimento dos requisitos;
- Documentação: inclui atividades de produção de documentos para o usuário final dos requisitos implementados e disponibilização dos mesmos;
- Testes: inclui as atividades de execução do controle da qualidade;
- Entrega: processo de disponibilização, parcial ou final, do *software* com os requisitos implementados.

Seção 2 – Documentos de Referência

São documentos de referência para este plano:

Documento	Versão	Data	Disponível em:
Processo de desenvolvimento da U.D. Projetos	001_2012	Maio/2012	Anexo A
Pesquisa de Atributos de Qualidade	001_2012	Maio/2012	
Plano de Verificação e Validação	001_2012	Maio/2012	
730 IEEE - Standard for <i>software</i> Quality Assurance Plans	2002	23/09/2002	

Seção 3 – Gestão

Esta seção apresenta os responsáveis pelas atividades de SQA e deste SQAP:

3.1 Organização

Pertencente à Área Técnica da empresa, a U.D. Projetos é responsável pela criação e desenvolvimento do *software* Promob e de produtos dele dependentes, por exemplo os plugins. Formada pelo seu gerente, desenvolvedores e analistas de desenvolvimento, documentador, analista de testes e testador, os dois últimos compõe uma sub unidade, chamada Núcleo de Testes. Cada colaborador da equipe possui responsabilidades para com a Garantia da Qualidade:

O **gerente da UD** é responsável:

- pela aprovação do planejamento das ações apresentadas pelo Núcleo de Testes;
- pelo planejamento das ações de qualidade do processo;
- por providenciar a resolução de não conformidades do processo;

O **gerente de produto** é responsável:

- por providenciar a resolução de não conformidades do produto.

O **documentador** é responsável:

- Traduzir os requisitos técnicos em uma linguagem que possa ser compreendida pelas demais equipes da Promob e por clientes;

O(s) **analista(s) de desenvolvimento** e **desenvolvedor(es)** são responsáveis:

- Identificar soluções para os requisitos do *software*;
- Implementar as soluções propostas nas versões/compilações do *software*;

O **analista de teste** e **testador** são responsáveis:

- pelo planejamento e execução de qualquer ação de qualidade de produto UD;
- pela identificação e reporte de não conformidades de produto e processo;
- pelo acompanhamento de resolução de não conformidades;
- pela avaliação da resolução de não conformidades;
- pelo desenvolvimento e manutenção do Plano de Garantia da Qualidade do Promob.

3.2 Tarefas

São tarefas de SQA a serem realizadas para um projeto de desenvolvimento do *software* Promob, na etapa de Planejamento:

- **Revisar plano de produto:** esta atividade deve ser realizada antes da liberação de cada executável e consiste em identificar se todas as solicitações atendidas foram relacionadas para garantir que as mesmas serão testadas, documentadas e divulgadas. A revisão ocorre através da comparação do documento project produzido com as solicitações que possui status de concluída no portal de solicitações.

São tarefas de SQA a serem realizadas para um projeto de desenvolvimento do *software* Promob, na etapa de Análise e Implementação:

- **Revisar análise de solicitação:** nesta atividade, executada a cada solicitação atendida, ou seja, requisito implementado, visa que o desenvolvedor revise a solução dada.
- **Revisar código fonte:** nesta revisão o produto de trabalho da implementação deve ser avaliado quanto a sua aderência a práticas adotadas pela UD, bem como o funcionamento da solução desenvolvida.

São tarefas de SQA a serem realizadas para um projeto de desenvolvimento do *software* Promob, na etapa de Documentação:

- **Revisar documento de requisito implementado:** este documento explica a necessidade e a solução dada para as alterações realizadas para a entrega deve ser criado um para cada solicitação listada no plano de produto.

São tarefas de SQA a serem realizadas para um projeto de desenvolvimento do *software* Promob, na etapa de Testes:

- **Revisar casos de testes:** os casos de testes são produzidos antes de cada etapa de testes e devem contemplar as solicitações atendidas no executável testável disponibilizado, sua revisão deve verificar a aderência aos atributos de qualidade do Promob.

São tarefas de SQA a serem realizadas para um projeto de desenvolvimento do *software* Promob, na etapa de Documentação:

- **Revisar documento de divulgação:** realizado na conclusão de cada projeto, este documento deve conter todas as solicitações listadas no plano de projeto, bem como o link para a documentação realizada para a mesma, sua revisão deve verificar a aderência a estes.

São tarefas de SQA a serem realizadas para um projeto de desenvolvimento do *software* Promob, na etapa de Disponibilização:

- **Revisar produto final de *software*:** realizado na conclusão de cada projeto, o produto agrega as solicitações atendidas no projeto em conclusão. A disponibilização requer o produto compilado e disponibilizado em diretório pré-definido.

São tarefas de SQA a serem realizadas ao final do projeto de desenvolvimento do *software* Promob:

- **Auditar a execução do processo:** esta auditoria deve ser executada na entrega do projeto realizada pela UD, consiste em verificar e todas as etapas foram realizadas produzindo os documentos devidos.

No quadro a seguir, são apresentadas as atividades relacionadas acima, com os respectivos documentos de entradas e saídas, bem como a etapa de desenvolvimento a qual a atividade está associada:

Etapa	Entrada	Tarefas SQA	Saída
Disponibilização (produto final)	Plano de produto	Revisar plano de produto	Plano de produto revisado
Análise e implementação	Executável testável	Revisar análise da solicitação	Código-fonte revisado
		Revisar código fonte	Produto testável
Documentação	Solicitações	Revisar documento de requisito implementado	Documentação revisada
Testes	Casos de Testes	Revisar os Casos de Testes	Casos de testes revisados
Disponibilização (produto beta e final)	Documento de divulgação	Revisar o documento de entrega	Documento de divulgação revisado
	Revisar produto final de <i>software</i>	Revisar o produto final de <i>software</i>	Executável disponibilizado revisado
Disponibilização (produto final)	Processo da UD-Projetos	Auditar a execução do processo	Produto entregue

3.3 Funções e responsabilidades

Os cargo e/ou papel dos participantes do ciclo de desenvolvimento Promob são:

- **Gerente da UD:** gerencia as pessoas e os projetos da U.D. Projetos;
- **Gerente do projeto:** gerencia o projeto;
- **Gerente de negócio:** gerencia o projeto do ponto de vista do negócio;
- **Gerente de configuração:** gerencia códigos fontes dos produtos, mantendo a integridade dos mesmos;
- **Analista de desenvolvimento:** responde tecnicamente pelos requisitos do projeto;
- **Desenvolvedor:** codifica os requisitos do projeto;
- **Documentador:** documenta os requisitos implementados;
- **Analista de testes:** planeja os testes no *software* produzido;
- **Testador:** executa testes no *software* produzido.

Gerentes, analistas, desenvolvedores, documentador e testadores, cada é responsável pelo(s) documento(s) do qual é autor. Analista de testes e testador, reconhecidos como a subunidade Núcleo de Testes, respondem pela execução da Garantia da Qualidade. A tabela a seguir apresenta os responsáveis para cada tarefa:

Tarefas	Responsáveis
Revisar plano de produto	Gerente da UD
Revisar análise de solicitação	Desenvolvedor responsável
Revisar código fonte	Desenvolvedor responsável
Revisar documento de requisito implementado	Documentador
Revisar os Casos de Testes	Analista de testes
Revisar o documento de entrega	Documentador
Revisar produto final de <i>software</i>	Analista de testes
Auditar a execução do processo	Responsável por SQA

3.4 Recursos para SQA

A U.D. Projetos trabalha com o orçamento de 10 a 15% do tempo de desenvolvimento do projeto dedicado a atividades de qualidade. Este percentual deve ser mantido, porém não será mais utilizado apenas pelo Núcleo de Testes, mas sim por toda a equipe, durante todo o ciclo de desenvolvimento do produto, conforme processo de SQA.

Seção 4 – Documentação

Esta seção tem como objetivos:

- Identificar a documentação que rege desenvolvimento, verificação e validação, uso e manutenção do *software*, e;
- Identificar como os documentos devem ser verificados quanto à adequação.

4.1 Documentação para V&V, uso e manutenção do *software*

Esta documentação rege a verificação e validação, uso e manutenção do *software*.

4.1.1 Da descrição de requisitos de *software*

Os documentos de requisitos dos projetos da U.D. Projetos são as solicitações do Portal de solicitações. Podem ser descritas pelo cliente e transcritas pela equipe de atendimento ou por outras equipes que possuem contato diretamente com o cliente. Solicitações da equipe Promob, como a equipe de customização, também são recebidas. As solicitações reportam informações referente a necessidade, e em alguns casos a sugestão de solução.

Solicitações - Novo Item

Editar

Salvar Fechar Colar Recortar Copiar

Confirmar Área de Transferência

Departamento	Núcleo de Testes
Serviço	Compilação
Título	Teste
Descrição	Realizando um teste
Interessados	Marcio - Promob ;
Anexos	Clique aqui para anexar um arquivo
	Padrão

4.1.2 Do plano de V & V (verificação e validação)

Para que os produtos de *software* cumpram o uso pretendido e as expectativas dos usuários faz-se uso do plano de verificação e de validação,

apresentado abaixo, que determinam se os produtos de *software* desenvolvidos estão conforme os requisitos, para que cumpram o uso pretendido e as expectativas dos usuários.

Plano	Entrega	Nível	Estratégias	Abordagem	Fase	Categoria	Método para obter CT	Responsáveis
Validação	Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Testes (Testes de Unidade)	Funcionalidade	Decomposição de requisitos	Desenvolvedor
	Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Testes (Testes de Unidade)	Usabilidade	Decomposição de requisitos	Desenvolvedor
	Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Testes (Testes de Unidade)	Segurança	Decomposição de requisitos	Desenvolvedor
	Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Aceitação)	Funcionalidade	Decomposição de requisitos	Analista de Testes/Testador
	Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Aceitação)	Usabilidade	Decomposição de requisitos	Analista de Testes/Testador
	Versão	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Aceitação)	Segurança	Decomposição de requisitos	Analista de Testes/Testador
	Versão	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Sistema)	Volume	Decomposição de requisitos	Analista de Testes/Testador
	Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Integração)	Compatibilidade	Decomposição de requisitos	Analista de Testes/Testador
	Versão	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Sistema)	Performance	Decomposição de requisitos	Analista de Testes/Testador
	Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Sistema)	Confiabilidade e Disponibilidade	Decomposição de requisitos	Analista de Testes/Testador
	Versão	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Sistema)	Recuperação	Decomposição de requisitos	Analista de Testes/Testador
	Versão	Alto	Caixa-preta	Baseado em requisitos	Testes (Teste de Sistema)	Contingência	Decomposição de requisitos	Analista de Testes/Testador
	Versão	Alto	Caixa-preta	NA	Testes (Teste de Sistema)	Estrutural	Grupos do sistema	Clientes convidados
Versão	Alto	Caixa-preta	NA	Testes (Teste de Integração)	Beta-teste	NA	Clientes convidados	
Verificação	Versão e Compilação	Alto	NA	NA	Análise e Implementação	Verificação do documento de solicitação	NA	Desenvolvedor
	Versão e Compilação	Alto	NA	NA	Análise e Implementação	Verificação do código fonte	NA	Desenvolvedor

4.1.3 Do relatório de V & V

Informações da execução dos planos de validação e verificação, bem como resultados das métricas, apresentadas na próxima seção, deverão ser identificados, analisados e

apresentados em números e gráficos em um único relatório. Este deve ser publicado no blog do Núcleo de Testes a cada conclusão de projeto.

Selecione o projeto:
Compilação 4

Período de desenvolvimento

De:

Até:

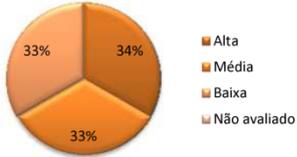
Data Entrega

Dia:

Hora:

REQUISITOS ATENDIDOS

Prioridade	Quantidade
Alta	1
Média	1
Baixa	1
Não avaliado	0
Total	4

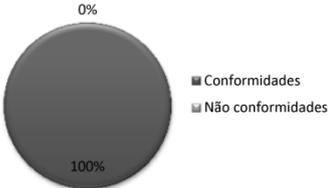


Categoria	Porcentagem
Alta	34%
Média	33%
Baixa	33%
Não avaliado	0%

RESULTADOS DE SQA



Status	Porcentagem
Verificações realizadas	50%
Verificações não realizadas	50%



Status	Porcentagem
Conformidades	100%
Não conformidades	0%

Etapa: PLANEJAMENTO	
Verificações realizadas	1
Verificações não realizadas	1
Conformidades	1
Não conformidades	0

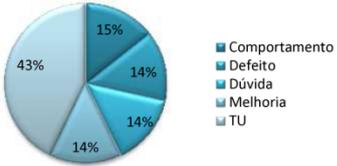
Etapa: ANÁLISE E IMPLEMENTAÇÃO	
Verificações realizadas	
Verificações não realizadas	
Conformidades	
Não conformidades	

Etapa: DOCUMENTAÇÃO	
Verificações realizadas	
Verificações não realizadas	
Conformidades	
Não conformidades	

Etapa: DISPONIBILIZAÇÃO	
Verificações realizadas	
Verificações não realizadas	
Conformidades	
Não conformidades	

Etapa: TESTES	
Verificações realizadas	
Verificações não realizadas	
Conformidades	
Não conformidades	

SEVERIDADE DOS ERROS	
Prioridade	Quantidade
Defeito	3
Melhoria	2
TU	1
Usabilidade	1
Padrão	0
Total	7
Solicitações atendidas	7
Solicitações pendentes	1



Categoria	Porcentagem
Comportamento	43%
Defeito	15%
Dúvida	14%
Melhoria	14%
TU	14%

4.1.4 Da documentação do usuário

O documento de requisito implementado descreve o requisito implementado para entendimento e uso do usuário final e de demais áreas da empresa. Após concluído é disponibilizado na base de conhecimento e segue o modelo

1. Quando ocorre?

Informa o momento em que a situação ocorre.

2. Por que ocorre?

Informa os motivos pelos quais a situação ocorre.

3. Como resolver?

Informa quais os procedimentos (passo a passo) que devem ser executados para resolução do problema ou direciona para links da Base de Conhecimento.

4. Links relacionados:

Informa links da Base de Conhecimento que estão relacionados à situação.

4.1.5 Do *software* plano de gerenciamento de configuração

Não há documento formalizando as práticas de gerenciamento de configurações, embora um padrão seja utilizado.

São *software* utilizados para a gerencia de configurações e que serão consideradas neste plano:

- **SVN** – Subversion - serviço utilizado para repositório de dados e para controle de versões de sistemas;
- **Forest**: Aplicação própria, cujo propósito é de manter um mapeamento entre as estruturas do cliente, onde as alterações são feitas, e do servidor SVN, bem como isolar as versões de produto testável e final entregues pela equipe de desenvolvimento.

São práticas adotadas para a gerencia de configurações e que serão consideradas neste plano:

- Após notificação da criação de um novo projeto de compilação ou versão, via e-mail, todos os desenvolvedores da equipe devem baixar os fontes, disponíveis no SVN, para desenvolvimento e remover os fontes do projeto anterior;
- Antes da edição dos códigos fontes, o arquivo deve ser bloqueado pelo desenvolvedor que aplica a alteração no SVN;

- Antes de submeter códigos fontes alterados, o desenvolvedor deve sincronizar sua base local com o SVN;
- Deve ser gerado a compilação dos componentes alterados (gerar os binários);
- Submeter fontes e componentes alterados e compilados ao SVN.

Os documentos gerados pelas atividade de SQA, que devem ser armazenados, devem seguir as práticas a seguir:

- Devem ser armazenados no servidor do Núcleo de testes, conforme políticas de acesso existentes. Um diretório, chamado SQA, deve ser criado e dentro deste um novo diretório cujo nome é o ano corrente (2012), este deverá conter um diretório para cada produto, nomeado com o nome do mesmo (Promob), seguindo padrão já utilizado neste servidor. Cada novo projeto, um novo diretório filho do produto deve ser criado, com o nome do projeto (compilacao2012) e neste serão depositados os arquivos de SQA, que não estiverem hospedados no servidor do *Sharepoint*.

4.1.5 Da disponibilização

O produto final de *software* deve ser disponibilizado no diretório pré-definido e a comunicação da mesma é realizada pelo blog da U.D. Projetos que fornece a lista de solicitações atendidas e o link para a base de conhecimento onde os documentos de requisito implementado estão publicados.

4.2 Produtos a serem revisados

A seguir a lista de documentos de produtos intermediários do *software* Promob a serem revisados, através da técnica determinada e conforme critérios descritos a aderencia deve ser confirmada. Os seguintes produtos e documentos estão envolvidos no processo SQA:

- **Plano de Projeto:** fornece a lista de solicitações que serão atendidas em um projeto de compilação ou versão. Permite gerenciar os recursos e o tempo durante o projeto.
- **Código fonte:** solução codificada computacionalmente correspondente a necessidade apresentada no requisito.
- **Casos de Testes:** descreve situações de testes específicos para validar a solução contra o requisito.
- **Documento de requisito implementado:** registra a necessidade e a solução dada para a solicitação, informa para quais distribuições se aplicam as implementações realizadas, descreve o que a solução permite e não permite

realizar e como usá-la. Para toda solicitação, ou requisito este documento é criado.

- **Produto final de *software*:** executável compilado do *software* Promob.
- **Documento de divulgação da entrega:** Comunica da disponibilização, fornece a lista de solicitações atendidas e o link para a base de conhecimento onde os documentos de requisito implementado estão hospedados.

Seção 5 - Normas, Práticas, Convenções e Métricas

As normas, práticas e processos adotados pela U.D. Projetos para o *software* Promob são:

5.1 Normas, práticas e Convenções

As normas, práticas e processos adotados pela U.D. Projetos para o *software* Promob são:

5.1.1 Do padrão de interface

A interface do Promob segue um padrão não formalizado, porém há consenso entre a equipe e interessados no atual padrão adotado. Sempre que houver necessidade de alterar a interface ou criar nova ferramenta, editor, etc deve ser analisado as já existentes para seguir a mesma padronização.

Já as mensagens emitidas para o usuário possuem um padrão formalizado que dever ser obrigatoriamente seguido, apresentado no anexo C deste plano.

5.2 Métricas

Nesta seção são descritas as métricas a serem aplicadas, a fim de medir a qualidade dos produtos e processo avaliados.

a) Métricas do Processo de Desenvolvimento

Seleção da Métrica	Descrição da métrica	Propósito	Fórmula
Retrabalho	Retrabalho	O retrabalho durante a etapa de desenvolvimento também deve ser medido, permitindo identificar o esforço que está sendo necessário devido aos defeitos não detectados nas etapas iniciais.	$\text{retrabalho} = (\text{Número de requisitos implementados}/\text{tempo de implementação}) / (\text{Número de requisitos implementados}/\text{tempo de correção})$.
Prioridade dos defeitos	Defeito de impacto baixo	A origem dos defeitos permite identificar os pontos mais frágeis nas etapas do processo e o volume é um indicador de confiança do produto.	$\text{Impacto} = (\text{número de solicitações do tipo "Defeito" e de priorização "Média" e priorização "Baixa"}) / \text{número total de defeitos}$.
	Defeito de impacto médio		$\text{Impacto} = (\text{número de solicitações do tipo "Defeito" e de priorização "Alto"}) / \text{número total de defeitos}$.
	Defeito de impacto alto		$\text{Impacto} = (\text{número de solicitações do tipo "Defeito" e de priorização "Urgente"}) / \text{número total de defeitos}$. Urgente: situações onde o software é finalizado durante a execução ou de alguma forma impede o cliente de dar seguimento a suas atividades através do software, tal como impossibilitar gerar qualquer tipo de saída.

b) Métricas do Processo de SQA

Processo	Seleção da Métrica	Descrição da métrica	Propósito	Fórmula
Processo de SQA	Distribuição dos erros por etapa	Planejamento	Erros identificados na etapa de planejamento	$\text{Erro} = \frac{\text{número de ocorrência "de não conformidades" na etapa de planejamento}}{\text{número total de requisitos}}$
		Análise e implementação	Erros identificados na etapa de análise e implementação	$\text{Erro} = \frac{\text{número de ocorrência "de não conformidades" na etapa de análise e implementação}}{\text{número total de requisitos}}$
		Documentação	Erros identificados na etapa de documentação	$\text{Erro} = \frac{\text{número de ocorrência "de não conformidades" na etapa de documentação}}{\text{número total de requisitos}}$
		Testes	Erros identificados na etapa de testes	$\text{Erro} = \frac{\text{número de ocorrência "de não conformidades" na etapa de testes}}{\text{número total de requisitos}}$
		Disponibilização	Erros identificados na etapa de disponibilização	$\text{Erro} = \frac{\text{número de ocorrência "de não conformidades" na etapa de disponibilização}}{\text{número total de requisitos}}$
	Testes	Severidade dos erros	Classificação dos erros conforme tipo de situação reportada	Erro de funcionalidade = número de ocorrência de erros na funcionalidade/número total de erros
				Erro de Padrão
Erro de Usabilidade				
Erro TU				
			$\text{Erro de Padrão} = \frac{\text{número de ocorrência de erros por falta de padrão}}{\text{número total de erros}}$	
			$\text{Erro de Usabilidade} = \frac{\text{número de ocorrência de erros de usabilidade}}{\text{número total de erros}}$	
			$\text{Erro TU} = \frac{\text{número de ocorrência de erros de TU}}{\text{número total de erros}}$	

c) Métricas do Produto Promob

As métricas de produto devem ser aplicadas aos projetos de versão. As relacionadas a característica eficiência e funcionalidade devem ser medidas pela equipe de testes. As métricas de usabilidade devem ser medidas pela equipe de documentação, com base nos relatórios da Central de Atendimento.

São métricas de produto relacionadas com a característica eficiência a serem adotadas:

Subcaracterísticas de Qualidade	Seleção da Métrica	Propósito	Fórmula
Comportamento de medidas de tempo	Tempo de resposta	Avaliar o tempo de resposta do sistema para completar a função.	$T = (\text{tempo de ganhar o resultado}) - (\text{Tempo de entrada de comando concluído})$
	Tempo de resposta médio		$X = T_{\text{mean}} / TX_{\text{mean}}$ $T_{\text{mean}} = \sum (Ti) / N$, (para $i = 1$ a N) $TX_{\text{mean}} = \text{necessário tempo médio de resposta}$ $Ti = \text{tempo de resposta para a } i\text{-ésima avaliação (tiro)}$ $N = \text{número de avaliações (tiros amostrados)}$
	Tempo de resposta pior		$X = T_{\text{max}} / R_{\text{max}}$ $T_{\text{max}} = \text{MAX } (Ti)$ (para $i = 1$ a N) $R_{\text{max}} = \text{tempo de resposta de no máximo}$ $\text{MAX } (Ti) = \text{tempo máximo de resposta entre avaliações}$ $N = \text{número de avaliações (tiros amostrados)}$ $Ti = \text{tempo de resposta para a } i\text{-ésima avaliação (tiro)}$
Comportamento de medidas de tempo	Tempo de espera	Medir o tempo de espera do usuário, após iniciar um grupo de tarefas relacionadas e sua conclusão.	$T = \text{Tempo entre usuário terminar ficando resultados de saída e solicitar acabamento usuário}$
	Tempo de espera médio		$X = T_{\text{mean}} / TX_{\text{mean}}$ $T_{\text{mean}} = \sum (Ti) / N$, (para $i = 1$ a N) $TX_{\text{mean}} = \text{necessário tempo médio de resposta}$ $Ti = \text{tempo de resposta para a } i\text{-ésima avaliação (tiro)}$ $N = \text{número de avaliações (tiros amostrados)}$
	Tempo de espera pior		$X = T_{\text{max}} / R_{\text{max}}$ $T_{\text{max}} = \text{MAX } (Ti)$ (para $i = 1$ a N) $R_{\text{max}} = \text{tempo de resposta exigido máxima}$ $\text{MAX } (Ti) = \text{tempo de resposta máximo entre avaliações}$ $N = \text{número de avaliações (tiros amostrados)}$ $Ti = \text{tempo de resposta para a } i\text{-ésima avaliação (tiro)}$

São métricas relacionadas a característica funcionalidade a serem adotadas:

Subcaracterísticas de Qualidade	Seleção da Métrica	Propósito	Fórmula
Adequação	Plenitude da implementação	Avaliar quão completa é a implementação conforme exigência especificações	$X = 1 - A / B$ $A = \text{Número de funções em falta detectada em avaliação}$ $B = \text{Número de funções descrito no especificações de requisitos}$
	Cobertura da implementação	Avaliar a corretude da implementação	$X = 1 - A / B$ $A = \text{Número de funções implementadas incorretamente ou funções em falta detectada na avaliação}$ $B = \text{Número de funções descrito no especificações de}$
Acurácia	Precisão	Avaliar quantas vezes os usuários encontram resultados com inadequado precisão.*	$X = A / T$ $A = \text{Número de resultados encontrados pelos usuários com o nível de precisão diferente exigido}$ $T = \text{tempo de operação}$
Segurança	Prevenção na corrupção de dados	Avaliar qual é a frequência de corrupção de projetos ou documentos de um projeto, através dos chamados de suporte realizar a avaliação dos dados.	a) $X = A / N$ $A = \text{Número de vezes que projetos ou documentos do projeto corrompeu}$ $N = \text{Número total de usuários ativos do Promob}$

São métricas relacionadas a característica usabilidade a serem adotadas:

Subcaracterísticas de Qualidade	Seleção da Métrica	Propósito	Fórmula
Compreensibilidade	Manifestação eficácia	Avaliar a proporção de funções que o usuário pode operar com sucesso, depois de uma demonstração ou tutorial.*	$X = A / B$ A = Número de funções operado com sucesso B = Número de demonstrações/tutoriais acessados
Aprendizabilidade	Facilidade de aprendizagem da função	Avaliar o tempo de duração para o usuário aprender a usar uma função.*	T = tempo médio necessário para aprender a usar um funcionar corretamente
	Acessibilidade da base de conhecimento	Avaliar a proporção de tópicos de ajuda que o usuário pode localizar.	$X = A / B$ A = Procedimentos pesquisados na base de conhecimento, sem retorno B = Total de número de procedimentos pesquisados
	Eficácia da base de conhecimento	Avaliar a proporção de tarefas que podem ser concluídas corretamente, após o usuário utilizar a documentação e/ou sistema de ajuda.	$X = A / B$ A = Número de chamados abertos com pedidos de ajuda para as funcionalidades do produto, concluídos com êxito, com apenas uma interação B = Total de chamados abertos com pedidos de ajuda para as funcionalidades do produto
Operacionalidade	Facilidade de correção de erros	Avaliar a facilidade do usuário na correção de erros nas tarefas.*	$T = T_c - T_s$ Tc = Tempo de completar correção de especificados erros de tipo de tarefa executada Ts = Tempo de correção a partir de determinado erros de tipo de tarefa executada
	Compreensibilidade de mensagem de erro	Avaliar a facilidade do usuário entender mensagens sistema de software.*	$X = A / UOT$ A = número de vezes que o usuário faz uma pausa por um longo período de tempo ou sucessivamente, e repetidamente falhar na mesma operação, devido à falta de compreensão da mensagem. UOT = tempo de operação durante período de observação
	Auto explicatividade de mensagens de erro	Avaliar a proporção de condições de erro que o usuário propõe a recuperação correta da ação.*	$X = A / B$ A = Número de condições de erro para o qual o usuário propõe a ação de recuperação correta B = Número de condições de erro testado
Atratividade	Atratividade da interface	Como atrativo é o interface para o usuário.*	Questionário para os usuários.

* São métricas aplicáveis através da adoção de testes alfa.

Seção 6 – Revisões

Esta seção define quais as revisões do *software* devem ser realizados, como devem ser aplicadas e o momento em que devem ser executadas, relacionando com o processo de desenvolvimento do projeto de *software*, conforme mostra a tabela abaixo.

Walkthrough é a técnica a ser utilizada, ela auxilia na detecção de falhas examinando o produto de *software* e a documentação pertinente através de um formulário, o *checklist*. A técnica de auditoria é uma revisão mais formal aplicada através de critérios objetivos. A auditoria a ser realizada avalia o processo e o produto de *software*, através da execução das atividades de SQA, buscando identificar a conformidade em relação aos padrões e procedimentos pré-estabelecidos.

Tarefas SQA	Técnica	Fase	Documento de Apoio
Revisar plano de produto	Walkthrough	A cada solicitação implementada	Checklist do plano de produto
Revisar análise	Walkthrough	A cada solicitação implementada	Checklist de análise
Revisar código fonte	Walkthrough	A cada solicitação implementada	Checklist de implementação
Revisar documento de requisito implementado	Walkthrough	A cada solicitação implementada	Checklist de documentação
Revisar os Casos de Testes	Walkthrough	A cada executável beta gerado	Checklist de Casos de Testes
Revisar produto final de software	Walkthrough	Após finalização da compilação/ versão (produto final)	Checklist de Entrega
Revisar o documento de divulgação	Walkthrough	Após finalização da compilação/ versão (produto final)	Checklist de entrega
Auditar a execução do processo	Auditoria	Após entrega compilação/ versão (produto final)	Checklist de Auditoria de SQA

Os *checklists* necessários para realizar as revisões de verificação e validação, de cada tarefa, foram estabelecidos, considerando as características de qualidade desejáveis para o produto. A elaboração dos mesmos foi realizada em conjunto com a equipe de desenvolvimento. Analistas que atuavam no processo ao qual o SQAP foi aplicado foram convidados a participar e em reunião discutiu-se cada um dos *checklists* e suas questões. Os *checklists* estão disponíveis no repositório de SQA da U.D. Projetos, conforme seção 4.

Seção 7 – Testes

Esta seção apresenta os testes não incluídos na verificação de *software*, quando houver, e um plano de validação para o *software* Promob.

Alguns testes são aplicados somente a determinados tipos de entrega (compilação ou versão) e, estão assim destacados na tabela abaixo.

Plano de Validação

Entrega	Nível	Estratégias	Abordagem	Fase	Categoria	Método para obter CT	Responsáveis
Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Teste de Unidade	Funcionalidade	Decomposição de requisitos	Desenvolvedor
Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Teste de Unidade	Usabilidade	Decomposição de requisitos	Desenvolvedor
Versão e Compilação	Baixo	Caixa-branca	Baseado em requisitos	Teste de Unidade	Segurança	Decomposição de requisitos	Desenvolvedor
Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Teste de Aceitação	Funcionalidade	Decomposição de requisitos	Analista de Testes/Testador
Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Teste de Aceitação	Usabilidade	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	Baseado em requisitos	Teste de Aceitação	Segurança	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	Baseado em requisitos	Teste de Sistema	Volume	Decomposição de requisitos	Analista de Testes/Testador
Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Teste de Integração	Compatibilidade	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	Baseado em requisitos	Teste de Sistema	Performance	Decomposição de requisitos	Analista de Testes/Testador
Versão e Compilação	Alto	Caixa-preta	Baseado em requisitos	Teste de Sistema	Confiabilidade e Disponibilidade	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	Baseado em requisitos	Teste de Sistema	Recuperação	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	Baseado em requisitos	Teste de Sistema	Contingência	Decomposição de requisitos	Analista de Testes/Testador
Versão	Alto	Caixa-preta	NA	Validação do Sistema	Estrutural	Grupos do sistema	Clientes convidados
Versão	Alto	Caixa-preta	NA	Validação do Sistema	Beta-teste	ND	Clientes convidados

Seção 8 - Relatório de Problemas e Ações Corretivas

O Portal de solicitações é a ferramenta a ser utilizada para reportar e gerenciar as não-conformidades, sejam elas identificadas com o *software* ou com as atividades de SQA. Após acessar o Portal de Solicitações, uma nova solicitação deve ser incluída. Na inclusão, o solicitante deve preencher:

- **Departamento** como “Núcleo de Testes”, que é o responsável pelas atividades de SQA.
- **Serviços** como **SQA Promob**.
- **Título** é um campo livre, mas deve receber o nome do projeto em avaliação/execução, por exemplo, “versão 2012”.
- **Descrição** define a situação não conforme, campo livre que permite ao solicitante relatar a situação identificada. E por fim, a solicitação deve ser salva.

A imagem mostra a interface de usuário para criar uma nova solicitação. O formulário contém os seguintes campos:

Departamento	Núcleo de Testes
Serviço	Compilação
Título	SQA Promob
Descrição	
Interessados	
Anexos	Clique aqui para anexar um arquivo

Na segunda etapa, o responsável pelo serviço selecionado recebe um e-mail informando do cadastro da solicitação que vai providenciar a correção. Para isso, o avaliador pode usar do campo **Interessado** e **Discussão** para dar andamento da correção/alteração.

A terceira e última é a alteração do **Status** da solicitação para “concluída”. Se a solução for identificada na nova etapa de testes a solicitação é finalizada, caso contrário a mesma é reincluída e reinicia o fluxo de solicitações.

Seção 9 – Ferramentas de Apoio

São ferramentas de apoio:

- Canal de Solicitações (MS *Sharepoint*): para visualização das solicitações do projeto. Também para cadastro de solicitações resultante da verificação e validação de produtos de *software* e de auditorias de SQA.
- Base de conhecimento (MS *Sharepoint*): para visualização da documentação de requisito implementado.
- Portal do Núcleo de Testes (MS *Sharepoint*): para publicar os relatórios.
- MS Excel: Para criação e manutenção de casos de testes e relatórios.
- Google Docs: para criação d *checklists*.

Seção 10 - Controle de mídia

Esta seção identifica o acesso ao produto de *software*, a armazenagem e a proteção de acesso não autorizado ou danos. Conforme política de empresa, controlada pela área de Infraestrutura de TI – Tecnologia da Informação – cada unidade de desenvolvimento tem acesso a seu servidor, onde o produto de *software* é disponibilizado. Demais tem somente acesso restrito, desta forma deve estar o diretório onde o produto é disponibilizado. Em caso de dano acidental ao produto, o mesmo deve ser recuperado através da aplicação Subversion, gerando o executável final novamente, danos maiores exigem a recuperação de backups, de responsabilidade da equipe de TI.

Seção 11 - Registros, Manutenção e Retenção

Esta seção identifica a documentação SQA que deve ser mantida, indicando os métodos e as instalações a serem usado para montar, arquivar, salvar e manter esta documentação, e designará o período de retenção.

Documento	Periodo	Diretório
Solicitações	Tempo indeterminado	Servidor Sharepoint
Checklists de revisão	Tempo indeterminado	Servidor da U.D. Projetos
Relatórios de V & V	Tempo indeterminado	Servidor da U.D. Projetos
Métricas	Tempo indeterminado	Servidor da U.D. Projetos

Seção 12 – Treinamento

Esta seção identifica as atividades de formação necessárias para atender às necessidades do SQAP.

- Apresentação do SQAP e dos conceitos de SQA
A apresentação deste documento deve ser realizada em reunião, juntamente com os envolvidos no(s) projeto(s) em que este será aplicado.

Seção 13 - Gestão de Riscos

Os riscos e contingências serão gerenciados pelo Núcleo de testes, e são de responsabilidade do gerente do projeto e do Núcleo de Testes. São riscos referentes ao plano de SQA do Promob:

Risco	Probabilidade	Gravidade	Contingência
Atraso ou não entrega dos requisitos para as atividades de testes.	Média	Média	Contatar com superior imediato.
Levantamento equivocado ou distorcido dos requisitos.	Alta	Alta	Contatar com gerente do projeto responsável.
Falta de conhecimento no contexto da implementação.	Média	Alta	Verificar documentos disponíveis
			Solicitar reunião com envolvidos com o negócio.
Afastamento temporário do analista/testador.	Baixa	Alta	Ajuste do cronograma do projeto.
Especificações funcionais insuficientes ou inexistentes	Média	Alta	Contatar com gerente responsável pelo projeto
			Solicitar reunião com envolvidos com o negócio.
			Ajuste do cronograma.
Não ter contato com os interessados	Média	Média	Contatar com superior imediato.
Atraso na entrega do projeto	Alta	Baixa	Contatar com gerente do projeto.
			Ajuste do cronograma.

Seção 14 - Glossário

Termo	Descrição
IEEE	<i>Institute of Electrical and Electronic Engineers</i> (Instituto de Eletrônica e Engenharia Elétrica)
SVN	<i>Subversion</i>
SQA	Garantia da Qualidade <i>software</i>
SPAQ	Plano da Garantia da Qualidade <i>software</i>

Seção 15 - Procedimento de Controle de Mudança do SQAP e Histórico

Para modificação e/ou atualização deste documento deve ser preenchido a tabela abaixo:

Data	Versão	Descrição	Responsável
