

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

Camila Hübner Brondani

**AVALIAÇÃO DA QUALIDADE DE ARTEFATOS EM LINHAS DE
PROCESSO DE SOFTWARE ADAPTADA**

Santa Maria, RS
2016

Camila Hübner Brondani

**AVALIAÇÃO DA QUALIDADE DE ARTEFATOS EM LINHAS DE PROCESSO DE
SOFTWARE ADAPTADA**

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Informática (PPGI), Área de Concentração em Computação Aplicada, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação.**

Orientadora: Prof^ª. Dr^ª. Lisandra Manzoni Fontoura

Santa Maria, RS
2016

Ficha catalográfica elaborada através do Programa de Geração Automática da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Brondani, Camila Hübner
Avaliação da Qualidade de Artefatos em Linhas de
Processo de Software Adaptada / Camila Hübner Brondani.-
2016.
135 p.; 30 cm

Orientadora: Lisandra Manzoni Fontoura
Dissertação (mestrado) - Universidade Federal de Santa
Maria, Centro de Tecnologia, Programa de Pós-Graduação em
Informática, RS, 2016

1. Qualidade de Software 2. Artefatos de Software 3.
Adaptação de Processos 4. Linhas de Processo de Software
I. Fontoura, Lisandra Manzoni II. Título.

© 2016

Todos os direitos autorais reservados a Camila Hübner Brondani. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

Endereço: Universidade Federal de Santa Maria, Cidade Universitária, “Prof. Mariano da Rocha Filho”, Av. Roraima, 1000, LaCA, Sala 388 Anexo B - CT, Bairro Camobi, CEP: 97105-900, E-mail: chbrondani@inf.ufsm.br

Camila Hübner Brondani

AVALIAÇÃO DA QUALIDADE DE ARTEFATOS EM LINHAS DE PROCESSO DE SOFTWARE ADAPTADA

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Informática (PPGI), Área de Concentração em Computação Aplicada, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

Aprovado em 23 de agosto de 2016:

Lisandra Manzoni Fontoura, Dra. (UFSM)
(Presidente/Orientadora)

Josiane Kroll, Dra. (PUCRS)

Gustavo Zanini Kantorski, Dr. (UFSM)

Santa Maria, RS
2016

AGRADECIMENTOS

À Deus, que protege, abençoa e ilumina o meu caminho todos os dias.

Aos meus pais, Milton e Isabel, obrigada por sempre acreditarem em mim, por me darem força, motivação e incentivo para alcançar este objetivo.

Às minhas irmãs, Karine e Isadora, que em vários momentos foram compreensivas com meu cansaço e às vezes até com meu humor.

Ao meu namorado, Otávio, por ser meu companheiro em todas as horas. Obrigada pela paciência, carinho, apoio, compreensão e amor.

À minha orientadora, Prof. Dr^a. Lisandra, que me acompanha há mais de cinco anos, obrigada por sempre estar disposta a me ajudar, pelas conversas, orientações, amizade e pela confiança. Saiba que você foi muito importante nessa caminhada e que uma parcela desta conquista eu devo a ti.

Aos amigos que conquistei durante o mestrado, em especial a Aline, Catherine, Darcielle, Denise, Cristiano, Henrique e Gelson, obrigada por compartilharem comigo vários momentos de risadas, conversas, conselhos, desesperos e amizade.

Aos demais amigos, familiares e professores, que sempre torceram por mim e que de alguma forma contribuíram para o meu crescimento.

À Fundação de Amparo à Pesquisa do Rio Grande do Sul (Fapergs) pelo apoio financeiro durante o mestrado.

“Lute com determinação, abrace a vida com paixão, perca com classe e vença com ousadia, porque o mundo pertence a quem se atreve e a vida é muito bela para ser insignificante.”

– Charles Chaplin

RESUMO

AVALIAÇÃO DA QUALIDADE DE ARTEFATOS EM LINHAS DE PROCESSO DE SOFTWARE ADAPTADA

AUTORA: CAMILA HÜBNER BRONDANI

ORIENTADORA: DRA. LISANDRA MANZONI FONTOURA

Na engenharia de software é necessário considerar indicadores relacionados a qualidade, tais como: esforço, produtividade, tempo e custo de desenvolvimento visando um efetivo controle do projeto. Esses indicadores são afetados negativamente quando artefatos defeituosos são produzidos. Nesse caso, o custo de retrabalho para correção desses defeitos aumenta em relação ao tempo de descoberta dos mesmos. Desta forma, iniciativas devem ser realizadas no sentido de encontrar os defeitos e corrigi-los tão logo sejam introduzidos. Existem diferentes normas e modelos que visam garantir e melhorar continuamente a qualidade de software. Dentre estes, destaca-se o CMMI (*Capability Maturity Model Integration*), que é um modelo de referência que contém melhores práticas para desenvolvimento e manutenção de produtos, com um enfoque voltado para a maturidade de processos de software. Além disso, para que o processo de software seja efetivo em um determinado projeto é necessário que este considere as peculiaridades do projeto em questão. Por isso, modelos de avaliação de qualidade de processos propõem que processos organizacionais sejam adaptados às necessidades de um determinado projeto e seja criado como resultado o processo específico para este. Sendo assim, este trabalho apresenta uma abordagem para avaliação da qualidade de artefatos gerados e/ou transformados pelas diversas atividades que compõem um processo de software definido usando Linhas de Processo de Software (LPrS). As atividades do processo adaptado são selecionadas para satisfazer as práticas do modelo de qualidade CMMI. Esta proposta é apoiada em um *framework* de qualidade, composto por um metamodelo, uma base de conhecimento elaborada a partir do CMMI, um processo de avaliação, além de uma ferramenta de apoio. O *framework* incorpora princípios de Modelos de Qualidade para a avaliação de produtos de software e tem como principal objetivo a instanciação de um plano para avaliação da qualidade de artefatos de software. Por fim, a proposta foi avaliada e validada por meio da realização de estudos de caso reais de empresas distintas, e envolveram cinco profissionais com relevante experiência em desenvolvimento de software. Portanto, foram gerados cinco processos adaptados de acordo com o contexto dos projetos e conforme as práticas do CMMI selecionadas, além da elaboração dos planos de qualidade para avaliação dos artefatos gerados pelas atividades que compõem o processo. Os especialistas analisaram a abordagem e consideraram positiva a proposta defendida nesse trabalho.

Palavras-chave: Qualidade de Software. Artefatos de Software. Adaptação de Processos. Linhas de Processo de Software.

ABSTRACT

QUALITY EVALUATION OF ARTIFACTS IN TAILORED SOFTWARE PROCESS LINES

AUTHOR: CAMILA HÜBNER BRONDANI
ADVISOR: PROF. DRA. LISANDRA MANZONI FONTOURA

In software engineering, it is necessary to consider indicators related to quality, such as: effort, productivity, time and cost of development in order to achieve an effective project control. These indicators are negatively affected when defective artifacts are produced. In this case, the cost of rework to correct defects increases in relation to the time of their discovery. Therefore, initiatives should be conducted in order to find these defects and correct them as soon as they are introduced. There are different norms and models designed to ensure and continuously improve the quality of software. Among these, the CMMI (Capability Maturity Model Integration) is a reference model that contains best practices for development and maintenance of products, with a focused approach to the maturity of software processes. In addition to the software process to be effective in a particular project it is necessary that it considers the peculiarities of the project in focus. Therefore, quality assessment process models propose that organizational processes are tailored to the needs of a particular project and is created as a result of the specific process for this. Thus, this paper presents an approach for assessing the quality of artifacts generated and/or transformed by the several activities that have a software process defined using Software Process Lines (SPrL). The activities of the tailored process are selected to satisfy the practices of the quality model CMMI. This proposal is supported by a quality framework, consisting of a metamodel, a knowledge base developed from the CMMI, an evaluation process and a support tool. The framework incorporates principles of quality models for the evaluation of software products and has as main objective the instantiation of a plan to evaluate the quality of software artifacts. Finally, the proposal was evaluated and validated through real case studies of different companies and involved five professionals with relevant experience in software development. So it was generated five tailored processes according to the context of the projects and as the CMMI practices selected, in addition to the development of quality plans for evaluation of the artifacts generated by the activities of the process. The experts analyzed the approach and considered positively the proposal of this work.

Keywords: Software Quality. Software Artifacts. Process Tailoring. Software Process Lines.

LISTA DE FIGURAS

Figura 1 – ISO/IEC 9126: (a) características e subcaracterísticas relacionadas às Qualidades Internas e Externas; (b) características relacionadas à Qualidade em Uso.....	29
Figura 2 – Qualidade no Ciclo de Vida	30
Figura 3 – Relacionamento entre as normas ISO/IEC 9126 e ISO/IEC 14598.....	32
Figura 4 – Visão geral da ISO/IEC 14598-1	33
Figura 5 – Componentes do Modelo CMMI	37
Figura 6 – Fases e Disciplinas do RUP	39
Figura 7 – Passos para Adaptação de Processos de Software	42
Figura 8 – Metamodelo de Qualidade	47
Figura 9 – Sistemática de Adaptação de Processos de Software.....	49
Figura 10 – Modelagem da Abordagem de Adaptação de Processos de Software com a Avaliação da Qualidade dos Artefatos	53
Figura 11 – <i>Quality Assessment of Artifacts in Process - Metamodel (QAPro-M)</i>	54
Figura 12 – Arquitetura definida para a disciplina de Requisitos e Configuração e Gerenciamento de Mudanças	59
Figura 13 – Atividade <i>Analyze the Problem</i> e elementos de processo.....	60
Figura 14 – Linha de Processo de Software definida para a disciplina de Requisitos e Configuração e Gerenciamento de Mudanças	62
Figura 15 – Instância do QAPro-M	65
Figura 16 – Tela Principal do Sistema com Elementos de Qualidade.....	68
Figura 17 – Exemplo de Cadastro do Elemento <i>Quality Goal</i>	68
Figura 18 – Exemplo de Listagem do Elemento <i>Quality Goal</i>	69
Figura 19 – Arquitetura <i>Requirements and Configuration & Change Management</i>	70
Figura 20 – Requisitos de Adaptação: CMMI.....	71
Figura 21 – Detalhes Iniciais do Plano de Avaliação de Qualidade de Artefato.....	72
Figura 22 – Exemplo de um Plano de Avaliação de Qualidade de Artefato	72
Figura 23 – Edição de um Plano de Avaliação de Qualidade de Artefato	73
Figura 24 – Contexto do Projeto SIS-ASTROS	80
Figura 25 – LPrS do Projeto SIS-ASTROS	82
Figura 26 – Requisitos de Adaptação selecionados para o Projeto da Empresa Dataprev.....	85
Figura 27 – Priorização e Seleção das Atividades para o Processo do Projeto Dataprev	85
Figura 28 – LPrS do Projeto da Empresa Dataprev.....	86
Figura 29 – LPrS do Projeto de Inovação em Automação Agrícola	89
Figura 30 – LPrS do Projeto de Ponto Eletrônico da UFSM	92
Figura 31 – LPrS do Projeto Sistema Irriga	95
Figura 32 – Grau de Experiência dos Especialistas.....	97

Figura 33 – Atividade <i>Understand Stakeholder Needs</i> e elementos de processo.....	114
Figura 34 – Atividade <i>Refine the System Definition</i> e elementos de processo	114
Figura 35 – Atividade <i>Define the System</i> e elementos de processo	115
Figura 36 – Atividade <i>Manage the Scope of the System</i> e elementos de processo	115
Figura 37 – Atividade <i>Manage Changing Requirements</i> e elementos de processo.....	116
Figura 38 – Atividade <i>Plan Project Configuration & Change Control</i> e elementos de processo.....	116
Figura 39 – Atividade <i>Manage Baselines & Releases</i> e elementos de processo	117
Figura 40 – Atividade <i>Create Project Configuration Management (CM) Environments</i> e elementos de processo.....	117
Figura 41 – Atividade <i>Monitor & Report Configuration Status</i> e elementos de processo	118
Figura 42 – Atividade <i>Change and Deliver Configuration Items</i> e elementos de processo..	118
Figura 43 – Atividade <i>Manage Change Requests</i> e elementos de processo	119

LISTA DE TABELAS

Tabela 1 – CMMI: Áreas de Processo e seus Níveis de Maturidade	36
Tabela 2 – Associação entre Práticas do CMMI com as Atividades e Artefatos do RUP	58
Tabela 3 – Plano de Qualidade: Artefato Pedido de Mudança.....	66
Tabela 4 – Dados do CMMI para as Práticas da Área de Processo Gerenciamento de Configuração	66
Tabela 5 – Comparação dos Trabalhos Relacionados	78
Tabela 6 – Contexto do Projeto da Empresa Dataprev.....	84
Tabela 7 – Contexto do Projeto de Inovação em Automação Agrícola	88
Tabela 8 – Contexto do Projeto de Ponto Eletrônico da UFSM.....	91
Tabela 9 – Contexto do Projeto Sistema Irriga.....	94
Tabela 10 – Associação da Área de Processo Gerenciamento de Requisitos	112
Tabela 11 – Associação da Área de Processo Gerenciamento de Configuração.....	113
Tabela 12 – Plano de Qualidade: Artefato Plano de Gerenciamento de Configuração.....	120
Tabela 13 – Dados do CMMI para as Práticas da Área de Processo Gerenciamento de Configuração.	120
Tabela 14 – Plano de Qualidade: Artefato Repositório do Projeto	121
Tabela 15 – Dados do CMMI para as Práticas da Área de Processo Gerenciamento de Configuração.	121
Tabela 16 – Plano de Qualidade: Artefato Registro de Auditoria de Configuração	122
Tabela 17 – Dados do CMMI para as Práticas da Área de Processo Gerenciamento de Configuração.	122
Tabela 18 – Plano de Qualidade: Artefato Ordem de Serviço	123
Tabela 19 – Dados do CMMI para as Práticas da Área de Processo Gerenciamento de Configuração.	123
Tabela 20 – Plano de Qualidade: Artefato Registro de Revisão	124
Tabela 21 – Plano de Qualidade: Artefato Atributos de Requisitos.....	124
Tabela 22 – Plano de Qualidade: Artefato Especificações Suplementares	125
Tabela 23 – Plano de Qualidade: Artefato Requisitos de Software	125
Tabela 24 – Plano de Qualidade: Artefato Especificação de Requisitos de Software	126
Tabela 25 – Plano de Qualidade: Artefato Solicitação dos <i>Stakeholders</i>	126
Tabela 26 – Plano de Qualidade: Artefato Plano de Gerenciamento de Requisitos.....	127
Tabela 27 – Plano de Qualidade: Artefato Resultados do Teste	127
Tabela 28 – Plano de Qualidade: Artefato <i>Workspace</i>	128
Tabela 29 – Plano de Qualidade: Artefato <i>Test Log</i>	128
Tabela 30 – Métodos de Avaliação e Métricas de acordo com a ISO/IEC 9126	129

LISTA DE ABREVIATURAS E SIGLAS

AHP	<i>Analytic Hierarchy Process</i>
BPMN	<i>Business Process Modeling Notation</i>
CMM	<i>Capability Maturity Model</i>
CMMI	<i>Capability Maturity Model Integration</i>
CQA	<i>Continuous Quality Assessment</i>
CQA-Meth	<i>Continuous Quality Assessment Methodology</i>
CQA-Tool	<i>Continuous Quality Assessment Tool</i>
GQM	<i>Goal Question Metric</i>
IEC	<i>International Electrotechnical Commission</i>
ISO	<i>International Organization for Standardization</i>
JPA	<i>Java Persistence API</i>
JSF	<i>Java Server Faces</i>
KLOC	Número de erros por mil linhas de código
LPtS	Linha de Processos de Software
LPS	Linhas de Produtos de Software
MfTPt	<i>Metamodel for Tailoring Process tool</i>
MPS.Br	Melhoria de Processo do Software Brasileiro
NCU	<i>Number of Use Cases per Class</i>
NUC	<i>Number of Classes per Use Case</i>
PEnSo	Pesquisa em Engenharia de Software
PMBOK	<i>Project Management Body of Knowledge</i>
QAPro-F	<i>Quality Assessment of Artifacts in Process - Framework</i>
QAPro-M	<i>Quality Assessment of Artifacts in Process - Metamodel</i>
QAPro System	<i>Quality Assessment of Artifacts in Process System</i>
RMC	<i>Rational Method Composer</i>
RUP	<i>Rational Unified Process</i>
SEI	<i>Software Engineering Institute</i>
SG	<i>Specific Goals</i>
SP	<i>Specific Practices</i>
SPICE	<i>Software Process Improvement and Capability dEtermination</i>
SQuaRE	<i>Software product QUALity Requirements and Evaluation</i>
TODIM	Tomada de Decisão Interativa Multicritério
TOPSIS	<i>Technique for Order Preference by Smilarity the Ideal Solution</i>
UML	<i>Unified Modeling Language</i>
XP	<i>eXtreme Programming</i>

SUMÁRIO

1	INTRODUÇÃO.....	23
1.1	DEFINIÇÃO DO PROBLEMA	24
1.2	OBJETIVOS DO TRABALHO	25
1.3	ORGANIZAÇÃO DA DISSERTAÇÃO.....	26
2	REFERENCIAL TEÓRICO	27
2.1	MODELOS DE QUALIDADE DE PRODUTO	27
2.2	MODELO DE AVALIAÇÃO DA QUALIDADE DE PRODUTO.....	31
2.3	MODELOS DE QUALIDADE DE PROCESSOS	34
2.4	PROCESSOS DE SOFTWARE	37
2.4.1	<i>Rational Unified Process (RUP)</i>	38
2.4.2	Adaptação de Processos de Software	40
2.4.3	Linhas de Processos de Software.....	42
3	CONTEXTO DA PESQUISA.....	45
3.1	UMA ABORDAGEM PARA AVALIAÇÃO DA QUALIDADE DE ARTEFATOS DE SOFTWARE	45
3.2	UMA SISTEMÁTICA BASEADA NO REÚSO DE ATIVIDADES PARA ADAPTAÇÃO DE PROCESSOS	49
4	<i>FRAMEWORK</i> PARA AVALIAÇÃO DA QUALIDADE DE ARTEFATOS	53
4.1	METAMODELO	53
4.2	BASE DE CONHECIMENTO CONSTRUÍDA A PARTIR DO MODELO DE QUALIDADE CMMI.....	56
4.2.1	Uso do CMMI como Requisito de Adaptação.....	56
4.2.2	Elaboração da Arquitetura de Processo baseada no RUP	58
4.2.3	Elaboração da Linha de Processo de Software	61
4.3	PROCESSO DE ELABORAÇÃO DOS PLANOS DE QUALIDADE	62
4.3.1	Definição de Métodos de Avaliação e Métricas.....	67
4.4	FERRAMENTA DE APOIO <i>QAPRO SYSTEM</i>	67
5	TRABALHOS RELACIONADOS.....	75
5.1	ANÁLISE DOS TRABALHOS RELACIONADOS	77
6	ESTUDOS DE CASO	79
6.1	ESTUDO DE CASO 1 – PROJETO SIS-ASTROS	80
6.1.1	Avaliação do Especialista do Projeto SIS-ASTROS.....	82
6.2	ESTUDO DE CASO 2 – PROJETO DA EMPRESA DATAPREV	84
6.2.1	Avaliação do Especialista do Projeto da Empresa Dataprev.....	87
6.3	ESTUDO DE CASO 3 – PROJETO DE INOVAÇÃO EM AUTOMAÇÃO AGRÍCOLA.....	88

6.3.1	Avaliação do Especialista do Projeto de Inovação em Automação Agrícola.....	90
6.4	ESTUDO DE CASO 4 – PROJETO PONTO ELETRÔNICO DA UFSM.....	91
6.4.1	Avaliação do Especialista do Projeto de Ponto Eletrônico da UFSM	93
6.5	ESTUDO DE CASO 5 – PROJETO SISTEMA IRRIGA	94
6.5.1	Avaliação do Especialista do Projeto Sistema Irriga	95
6.6	ANÁLISE DOS ESTUDOS DE CASO.....	96
7	CONSIDERAÇÕES FINAIS	101
7.1	CONTRIBUIÇÕES	102
7.2	TRABALHOS FUTUROS.....	103
7.3	PUBLICAÇÕES	103
	REFERÊNCIAS	105
	APÊNDICE A – ASSOCIAÇÃO ENTRE CMMI E RUP	112
	APÊNDICE B – ATIVIDADES E ELEMENTOS DE PROCESSO	114
	APÊNDICE C – PLANOS DE QUALIDADE.....	120
	APÊNDICE D – EXEMPLOS DE MÉTODOS DE AVALIAÇÃO E MÉTRICAS.....	129
	APÊNDICE E – ETAPAS DO ESTUDO DE CASO	132

1 INTRODUÇÃO

Há um consenso que a adoção de processos de desenvolvimento de software exerce forte influência na qualidade final do software. Isso se deve, principalmente, pelo fato que o gerenciamento e a busca pela melhoria contínua da qualidade tendem a gerar software com menos defeitos e dentro dos padrões esperados. Existem diferentes normas e modelos para avaliação da qualidade de software, tais como ISO/IEC 12207 (ISO/IEC 12207, 2008), ISO/IEC 15504 (ISO/IEC 15504, 2003), MPS.Br (Melhoria de Processo do Software Brasileiro) (SOFTEX, 2011) e CMMI (*Capability Maturity Model Integration*) (SEI, 2010). Essas normas e modelos visam garantir e melhorar continuamente a qualidade de software. Além disso, para que o processo de software seja efetivo em um determinado projeto é necessário que este considere as peculiaridades do projeto em questão. Por isso, modelos de avaliação de qualidade de processos propõem que processos organizacionais sejam adaptados às necessidades de um determinado projeto e seja criado como resultado o processo específico para este (SEI, 2010).

As atividades de garantia de qualidade visam avaliar a qualidade dos artefatos produzidos em cada etapa do desenvolvimento, evitando que erros se propaguem. Al-Kilidar *et al.* (AL-KILIDAR; COX; KITCHENHAM, 2005) afirmam que ao invés de se tentar medir a qualidade de software como um todo, deve-se buscar a avaliação de atributos que compõem o produto que, quando combinados, podem fornecer uma noção geral sobre a sua qualidade. Contudo, embora qualidade seja um assunto recorrente em Engenharia de Software, as empresas – especialmente aquelas de pequeno e médio porte – carecem de especialistas capazes de definir as características de qualidade para os seus produtos (GUERRA; COLOMBO, 2009). Além disso, a definição e a classificação das características de qualidade, por si só, não são suficientes sem uma discussão de quais os meios necessários para alcançá-las e quais são os papéis interessados (MOHAGHEGHI; DEHLEN; NEPLE, 2008).

Visando padronizar a avaliação de produtos de software, tanto o produto final como dos diferentes artefatos que são produzidos durante o desenvolvimento, podem ser utilizados Modelos de Qualidade. Modelos de Qualidade têm sido aceitos para definir a qualidade em produtos de software, uma vez que buscam estruturá-la em fatores pontuais e fáceis de serem avaliados, ao mesmo tempo em que fornecem uma boa caracterização desses fatores (DEISSENBOECK *et al.*, 2009). Algumas referências desses modelos são descritas nos

trabalhos de Boehm *et al.* (BOEHM *et al.*, 1978), McCall *et al.* (MCCALL; RICHARDS; WALTERS, 1977) e pela norma ISO/IEC 9126 (ISO/IEC 9126, 2003).

Neste sentido, este trabalho propõe uma abordagem sistemática para a definição de um *framework* para a avaliação da qualidade de artefatos gerados e/ou transformados pelas atividades que compõem um processo de software adaptado. O *framework* é composto por um metamodelo, uma base de conhecimento elaborada a partir do modelo de qualidade CMMI, um processo de avaliação, cujo objetivo é organizar os conceitos que envolvem a definição de metas de qualidade e seus respectivos métodos e métricas de avaliação, além de uma ferramenta de apoio.

Em vista disso, o engenheiro de processos seleciona um conjunto de práticas do CMMI que precisa ser satisfeito no referido projeto. Então, as atividades que visam implementar essas práticas em um processo são recuperadas e inseridas no processo específico para o projeto. Cada atividade possui um conjunto de artefatos que são avaliados a partir de um conjunto de elementos. Estes são descritos em um metamodelo de qualidade que relaciona o processo de avaliação às diversas características que envolvem os artefatos, tais como seus propósitos, interessados, métodos e métricas correspondentes. O resultado desta proposta é a definição de planos para avaliação da qualidade dos artefatos do processo de software adaptado.

1.1 DEFINIÇÃO DO PROBLEMA

Um dos erros mais comuns sobre qualidade de software é acreditar que dentro do ciclo de desenvolvimento do software existe uma etapa alocada especialmente para as realizações de testes (BARTIÉ, 2002). Dessa forma, surge um equívoco, pois a qualidade não é obtida somente após a codificação de partes do produto a ser desenvolvido.

Não há dúvida de que o processo de desenvolvimento usado exerce uma significativa influência sobre a qualidade final do produto e que bons processos de software tendem a conduzir à construção de software de boa qualidade (SOMMERVILLE, 2011). (BERNABÉ *et al.*, 2011) afirmam que o sucesso de um projeto de desenvolvimento de software é principalmente dependente da qualidade dos artefatos produzidos em cada etapa do desenvolvimento.

Outro ponto importante a destacar é que o modelo de qualidade CMMI especifica *o que* deve ser feito e não *como* fazê-lo, então este trabalho busca contribuir com as

organizações que desejam atingir níveis de maturidade, sugerindo quais atividades devem compor o processo de desenvolvimento para satisfazer as práticas do CMMI.

Sendo assim, a questão de pesquisa definida para esta dissertação é *como avaliar a qualidade de artefatos gerados e/ou transformados pelas atividades que compõem um processo de software adaptado, desde os estágios iniciais do processo até o seu encerramento, com o propósito de melhorar a qualidade do produto final?*

1.2 OBJETIVOS DO TRABALHO

Nesta dissertação definiu-se a proposta de um *framework* de qualidade, cujo objetivo geral é avaliar a qualidade dos artefatos de software gerados ao longo de um processo de desenvolvimento de software adaptado.

Em vista disso, os objetivos específicos deste trabalho incluem:

- Elaboração de um metamodelo com o objetivo de estruturar tanto os elementos de processos, quanto os elementos de qualidade;
- Identificação e relacionamento entre as práticas específicas do CMMI com os artefatos e as atividades do RUP. Como o CMMI será um requisito de adaptação, o usuário irá selecionar as práticas desejadas e atividades que satisfaçam essas práticas serão buscadas no repositório a fim de formar o processo adaptado de acordo com a meta que deseja alcançar de uma determinada área de processo do CMMI;
- Elaboração de uma base de conhecimento de métodos de avaliação e métricas;
- Relacionamento entre os métodos de avaliação com as submetas de qualidade a partir do modelo de qualidade CMMI;
- Desenvolvimento de planos de qualidade a fim de identificar métodos e métricas mais apropriadas para avaliar artefatos gerados durante o processo de software adaptado por meio de Linha de Processos de Software (LPrS);
- Desenvolvimento do *QAPro System* a partir de uma extensão da ferramenta MfTPt¹.

¹ MfTPt refere-se a uma ferramenta *Web* que foi desenvolvida para apoiar a adaptação de processos de software. Esta ferramenta será descrita em detalhes na Seção 3.2.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Este trabalho está organizado como segue: no Capítulo 2, os conceitos importantes para a compreensão deste trabalho são apresentados. No Capítulo 3 é descrito o contexto da pesquisa no qual este trabalho está inserido. No Capítulo 4 é apresentado o *framework* proposto neste trabalho, que inclui: a) um metamodelo para avaliação de artefatos em processos de software adaptados, b) a criação de uma base de conhecimento a partir do modelo de qualidade CMMI, c) um processo de elaboração dos planos de qualidade dos artefatos e d) uma ferramenta de apoio denominada *QAPro System*. No Capítulo 5 trabalhos relacionados são discutidos. No Capítulo 6 são apresentados cinco estudos de caso para validar o *framework* proposto. Por fim, no Capítulo 7 as considerações finais e alguns passos futuros deste trabalho são discutidos.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico necessário para o entendimento desta dissertação. Na Seção 2.1 são apresentados os principais modelos de qualidade citados na literatura, tais como: o modelo de qualidade de McCall, o de Boehm e a ISO/IEC 9126. Na Seção 2.2 é descrito o modelo de avaliação da qualidade de produtos conhecida como ISO/IEC 14598. Na Seção 2.3 o modelo de qualidade de processos CMMI é detalhado. Na Seção 2.4 são descritos alguns conceitos básicos sobre o processo planejado *Rational Unified Process* (RUP), adaptação de processos e Linhas de Processos de Software (LPrS).

2.1 MODELOS DE QUALIDADE DE PRODUTO

Alguns modelos de qualidade servem como referência para descrever, classificar e organizar as características funcionais e não-funcionais de produtos (artefatos), estas são sistematicamente medidas e avaliadas. Estes modelos de qualidade são, de uma forma geral, classificados como Modelos Hierárquicos, Modelos Conceituais e Modelos de Contextos.

Para este trabalho foi utilizado os Modelos de Qualidade Hierárquicos, os quais descrevem a relação entre um conjunto fixo de fatores de qualidade de alto nível, atributos do produto e métricas apropriadas para alcançar esses fatores (TRENDOWICZ; PUNTER, 2003). Estes são organizados em pilares que são decompostos e refinados em atributos de qualidade específicos, capazes de serem avaliados quantitativamente por meio de métricas apropriadas. Os modelos hierárquicos mais relevantes a esta pesquisa e que, ao mesmo tempo, serviram como base teórica para o desenvolvimento de outros modelos mais complexos e elaborados, além de normas e padrões internacionais de qualidade são: o modelo de qualidade de McCall (MCCALL; RICHARDS; WALTERS, 1977), o de Boehm (BOEHM et al., 1978) e a ISO/IEC 9126 (ISO/IEC 9126, 2003).

Um dos primeiros modelos hierárquicos foi proposto por McCall *et al.* (MCCALL; RICHARDS; WALTERS, 1977). Este modelo teve origem nos meios militares americanos e foi utilizado, primeiramente, pelos seus desenvolvedores, para o processo do desenvolvimento de seus sistemas. O modelo visa estabelecer uma ligação entre usuários e desenvolvedores, focalizando nos fatores de qualidade do software que refletem as opiniões dos usuários e as prioridades dos desenvolvedores (GUERRA; COLOMBO, 2009).

É composto por três perspectivas principais para definir e identificar a qualidade de um produto de software: revisão do produto (habilidade de se submeter a mudanças), transição do produto (adaptabilidade a novos ambientes operacionais ou de *hardware*) e operação do produto (características básicas de operação do produto). McCall *et al.* definem 11 fatores de qualidade, são eles: *manutenibilidade*, *flexibilidade*, *testabilidade*, *portabilidade*, *reusabilidade*, *interoperabilidade*, *correção*, *confiabilidade*, *eficiência*, *integridade* e *usabilidade*. Cada um dos fatores subdivide-se em 23 critérios de qualidade, que ainda podem ser subdivididos em medidas de controle, as quais não estão definidas nesse modelo.

Os fatores de qualidade descrevem tipos diferentes de características comportamentais do software, e os critérios de qualidade são atributos a um ou mais dos fatores de qualidade. A ideia do modelo de qualidade de McCall é que os fatores de qualidade sintetizados forneçam um retrato completo da qualidade do software (GUERRA; COLOMBO, 2009).

O modelo de Boehm *et al.* (BOEHM; BROWN; LIPOW, 1976)(BOEHM et al., 1978) surgiu em 1978 e é similar ao de McCall. É um modelo hierárquico de qualidade, estruturado em três níveis, em que as características de alto nível representam as expectativas dos usuários em relação à avaliação da qualidade do software. Essas características são classificadas em *utilidades gerais* que incluem *portabilidade* e *manutenibilidade*, e características que os autores consideraram essenciais, porém não suficientes (*as-is utilities*).

As características de nível intermediário representam sete fatores de qualidade, decompostos das características de alto nível, que juntos descrevem as qualidades esperadas para um sistema de software. As *as-is utilities* são decompostas em *confiabilidade*, *eficiência* e *engenharia humana*. Manutenibilidade é decomposta em *testabilidade*, *compreensibilidade* e *flexibilidade*, e outro fator deste nível é *portabilidade*. Por fim, as características primitivas fornecem a fundamentação para definir as métricas de qualidade relacionadas a um ou mais dos fatores descritos no nível anterior (GUERRA; COLOMBO, 2009).

Embora os modelos de McCall e de Boehm possam parecer muito similares, existem diferenças com relação à hierarquia das características. Entretanto, pode-se afirmar que foram esses modelos que inspiraram os atuais modelos de qualidade.

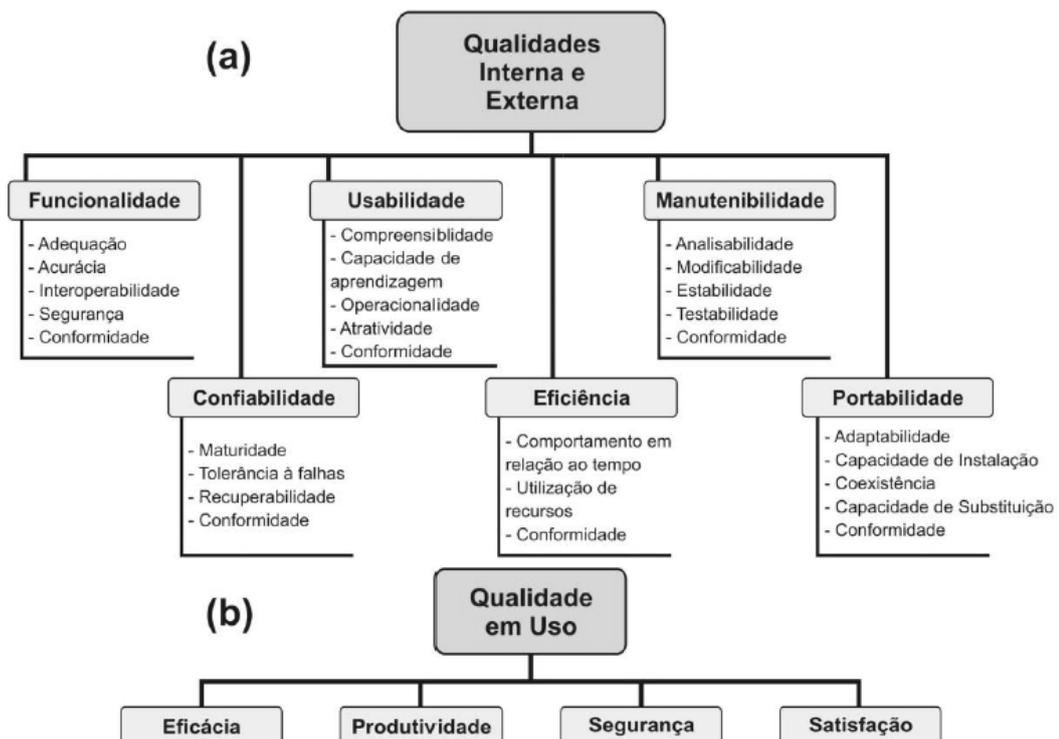
Por fim, proposta em 1991, a ISO/IEC 9126 (ISO/IEC 9126, 2003) foi posteriormente revisada e estabelecida como padrão internacional para avaliação da qualidade de produtos de software em 2001 (AL-KILIDAR; COX; KITCHENHAM, 2005). A norma ISO/IEC 9126 está preocupada em garantir que as necessidades dos clientes, em relação ao produto de software, sejam providas. Esta norma define quais são as características que um produto de

software deve ter e fornece um modelo para ser utilizado em uma avaliação de verificação da presença de tais características (GUERRA; COLOMBO, 2009).

O modelo de qualidade da ISO/IEC 9126 é formado por quatro partes, que são (ISO/IEC 9126, 2003):

1. ISO/IEC 9126-1: descreve o *framework* do modelo de qualidade. A primeira parte desse modelo especifica seis amplas categorias de características de qualidade dos produtos de software: *funcionalidade*, *confiabilidade*, *usabilidade*, *eficiência*, *manutenibilidade* e *portabilidade*. Estas, por sua vez, são divididas em subcaracterísticas, conforme mostrado na Figura 1;
2. ISO/IEC 9126-2: determina a qualidade externa do software;
3. ISO/IEC 9126-3: determina a qualidade interna de software;
4. ISO/IEC 9126-4: descreve a qualidade em uso. Esta parte é a visão de qualidade sob a perspectiva do usuário e é medida pelo efeito do uso do software. Este modelo faz a avaliação de quanto o usuário pode atingir seus objetivos em um ambiente, sem medir as propriedades do produto de software. O modelo de qualidade em uso possui quatro características, são elas: *efetividade*, *produtividade*, *segurança* e *satisfação*.

Figura 1 – ISO/IEC 9126: (a) características e subcaracterísticas relacionadas às Qualidades Internas e Externas; (b) características relacionadas à Qualidade em Uso.



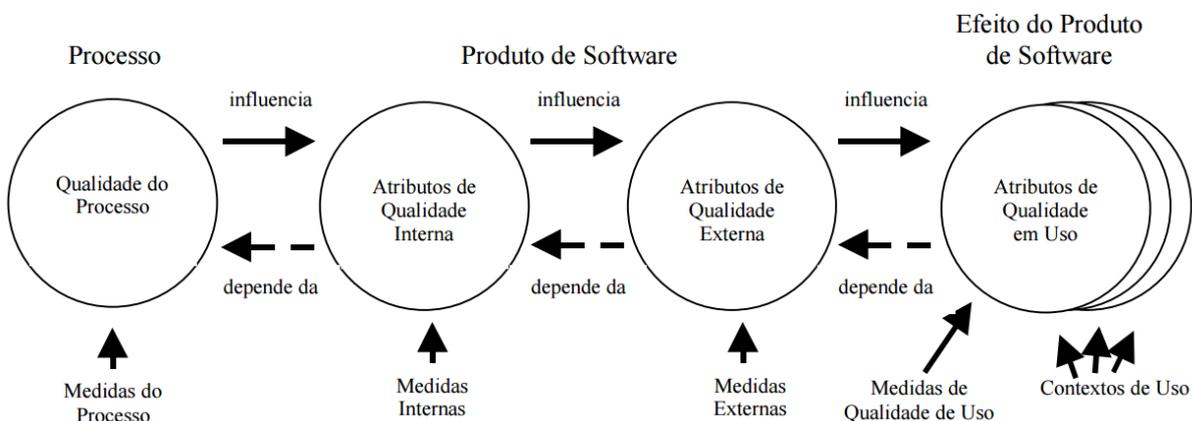
Sabe-se, no entanto, que as características da qualidade definidas por essa norma não são diretamente mensuráveis. É necessário um desdobramento das características em níveis mais específicos, até chegar a um ponto em que se consiga obter uma medida objetiva (GUERRA; COLOMBO, 2009). Vanderose (VANDEROSE, 2012) afirma que o modelo de qualidade da ISO/IEC 9126 apresenta a desvantagem de não mostrar claramente como as características podem ser medidas, dado que não há um processo de avaliação definido na norma.

Desta maneira, para saber o valor de uma determinada característica ou subcaracterística de qualidade, deve-se criar uma métrica para quantificá-la e assim, fazer uma medição. As métricas devem resultar em um valor matemático que verifique se o produto de software tem qualidade. Sob o ponto de vista de medição, as métricas de software podem ser de dois tipos: diretas ou indiretas.

As métricas diretas são resultantes de atributos observados, normalmente determinados por contagem, como o custo, número de linhas de código, número de páginas e capacidade de memória, entre outros. As métricas indiretas são obtidas a partir de métricas diretas, e entre seus exemplos estão funcionalidade, confiabilidade, eficiência e complexidade (PRESSMAN, 2011). Em vista disso, as métricas usadas pela norma ISO/IEC 9126 são consideradas métricas indiretas e são classificadas em três tipos: métricas externas, internas e de qualidade de uso, como já descrito.

A Figura 2 mostra a qualidade no ciclo de vida, no qual para obter qualidade em uso é preciso ter qualidade externa, que por sua vez é dependente de qualidade interna.

Figura 2 – Qualidade no Ciclo de Vida



Fonte: (ISO/IEC 9126, 2003).

Qualidade interna e externa são aplicáveis ao produto de software e a qualidade em uso é aplicável ao efeito do produto de software em um cenário específico. As métricas internas podem ser aplicadas a um produto de software não executável, enquanto as métricas externas podem ser usadas para medir a qualidade do produto de software por meio da medição de seu comportamento em um sistema do qual ele faça parte. E por fim, as métricas de qualidade em uso medem o quanto o produto agrega às necessidades de usuários específicos.

A ISO/IEC 9126 foi atualizada e integrada à série ISO/IEC 25000 (ISO/IEC 25000, 2005), também conhecida como *Software product QUALity Requirements and Evaluation (SQuaRE)*. Essa nova geração do padrão tem por objetivo satisfazer às crescentes necessidades dos usuários por meio de um conjunto melhorado e unificado de documentos normativos que abrangem três processos complementares de qualidade: especificação de requisitos, medição e avaliação (SURYN; ABRAN, 2003), além de incluir o padrão ISO/IEC 14598 (ISO/IEC 14598, 1999) como processo de avaliação das características de qualidade descritas pela ISO/IEC 9126.

2.2 MODELO DE AVALIAÇÃO DA QUALIDADE DE PRODUTO

Além de explorar o tema qualidade de produto de software, surge a necessidade de estudar como implementar e aplicar um processo de avaliação da qualidade de produto de software, com o objetivo de verificar a qualidade deste em relação a requisitos de padrões reconhecidos internacionalmente.

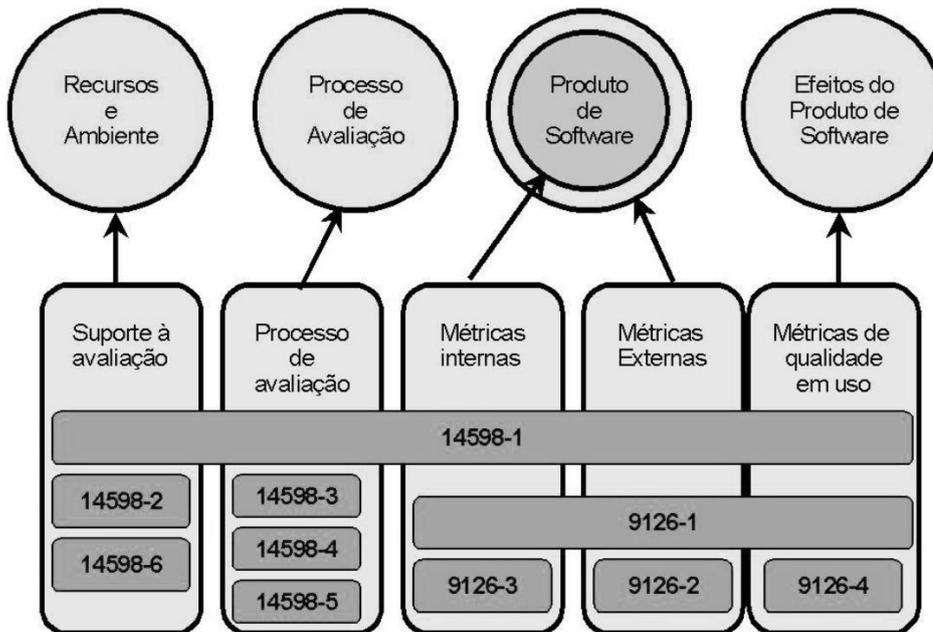
Deste modo, a norma ISO/IEC 14598 (ISO/IEC 14598, 1999) fornece requisitos e recomendações para implementação prática da avaliação de qualidade de produto de software. O processo de avaliação proposto pode ser usado para avaliar produtos já existentes ou produtos intermediários, isto é, em desenvolvimento. Esta pode ser utilizada por laboratórios de avaliação, fornecedores de software, compradores de software, usuários e entidades certificadoras, cada qual com seu objetivo (GUERRA; COLOMBO, 2009). A elaboração da série de normas ISO/IEC 14598 consolidou-se em seis partes:

1. ISO/IEC 14598-1: Visão Geral;
2. ISO/IEC 14598-2: Planejamento e Gerenciamento;
3. ISO/IEC 14598-3: Processo para a equipe de desenvolvimento;
4. ISO/IEC 14598-4: Processo para o comprador;
5. ISO/IEC 14598-5: Processo para o avaliador;

6. ISO/IEC 14598-6: Módulos de Avaliação.

As normas das séries ISO/IEC 9126 e ISO/IEC 14598 podem ser utilizadas em complementação, umas às outras, de acordo com o objetivo da avaliação. O relacionamento entre estas normas está ilustrado na Figura 3.

Figura 3 – Relacionamento entre as normas ISO/IEC 9126 e ISO/IEC 14598



Fonte: (ISO/IEC 14598, 1999).

A ISO/IEC 14598-1 contém conceitos de como avaliar a qualidade de software e define um modelo de processo de avaliação genérico. As normas ISO/IEC 14598-2 e ISO/IEC 14598-6 estabelecem itens necessários para o suporte à avaliação. Já as normas ISO/IEC 14598-3, ISO/IEC 14598-4 e ISO/IEC 14598-5, estabelecem processo de avaliação específico para desenvolvedores, adquirentes e avaliadores de software, respectivamente (GUERRA; COLOMBO, 2009).

A ISO/IEC 14598 afirma que a subjetividade da avaliação deve ser mínima, e para isso espera-se que o processo de avaliação tenha como características fundamentais: ser repetível, reproduzível, imparcial e objetivo. O processo de avaliação proposto pela norma ISO/IEC 14598-1 inclui quatro etapas de avaliação, contendo dez atividades, como mostrado na Figura 4.

Figura 4 – Visão geral da ISO/IEC 14598-1



Fonte: (ISO/IEC 14598, 1999).

Cada etapa apresenta passos bem determinados, que podem garantir o resultado esperado do processo. As etapas definidas pela ISO/IEC 14598 são (ISO/IEC 14598, 1999):

- Estabelecer requisitos de avaliação: o objetivo desta fase é estabelecer o que se quer avaliar e o que se quer como resultado da avaliação. O requisitante e o coordenador da avaliação têm atuação essencial nesta etapa, para que, juntos, possam consolidar os requisitos da avaliação e esclarecer o resultado a ser obtido. Esta etapa divide-se em três passos: a) estabelecer o propósito da avaliação; b) identificar tipos de produtos a serem avaliados; e c) especificar o modelo de qualidade.
- Especificar a avaliação: o objetivo desta fase é definir as medidas a serem utilizadas na avaliação e estabelecer suas respectivas pontuações, para serem representadas como resultado da avaliação. O coordenador da avaliação e o profissional com conhecimentos estatísticos têm atuação essencial nesta fase, pois será necessário converter as medidas obtidas numa escala numérica normalizada. Divide-se em três passos: a) selecionar métricas; b) estabelecer níveis de pontuação para as métricas; e c) estabelecer critérios para julgamento.
- Projetar a avaliação: o único passo desta etapa é produzir o plano de avaliação. Esse plano informa qual método de avaliação será utilizado, com instruções de como utilizá-lo, e especifica os recursos necessários, juntamente com o cronograma das ações para o avaliador.

- Executar a avaliação: nesta fase do processo de avaliação, o avaliador utilizará tudo o que foi preparado nas fases anteriores. Consiste na inspeção, medição e teste dos produtos e seus componentes de acordo com o plano de avaliação. A quarta etapa divide-se em três passos: a) obter as medidas; b) comparar com critérios; e c) julgar os resultados.

2.3 MODELOS DE QUALIDADE DE PROCESSOS

Para o sucesso na definição e melhoria dos processos de software, é fundamental que vários aspectos sejam considerados. Alguns modelos e normas de qualidade de processo, tais como ISO/IEC 9000-3 (ISO/IEC 9000, 2005), ISO/IEC 12207 (ISO/IEC 12207, 2008), ISO/IEC15504 (ISO/IEC 15504, 2003), Melhoria de Processo do Software Brasileiro (MPS.Br) (SOFTEX, 2011) e *Capability Maturity Model Integration* (CMMI) (SEI, 2010), surgiram com o objetivo de apoiar a busca por processos de melhor qualidade, apontando os principais aspectos que um processo de qualidade deve considerar.

A série de normas ISO/IEC 9001 foi desenvolvida para aplicação em qualquer setor produtivo. Para facilitar sua aplicação para desenvolvimento, suporte e manutenção de software, a ISO desenvolveu o guia ISO/IEC 9000-3 (ISO/IEC 9000, 2005). Outra norma da ISO para aplicação em desenvolvimento de software é a ISO/IEC 12207 (ISO/IEC 12207, 2008), que trata dos processos de ciclo de vida de software. A abordagem dessas normas da série ISO é fundamentada nos preceitos da documentação do sistema de qualidade que estabelece a visão da empresa com relação aos interesses e necessidades dos clientes e por isso resulta na percepção desses. A norma ISO/IEC 15504 (ISO/IEC 15504, 2003) foi desenvolvida em conjunto com a comunidade internacional por meio do projeto SPICE (*Software Process Improvement and Capability dEtermination*) com base em modelos já existentes como ISO/IEC 9000 e CMM.

O MPS.Br (SOFTEX, 2011) é um modelo ajustado à realidade brasileira de desenvolvimento de software, elaborado para fomentar a qualidade de software em pequenas e médias empresas. O modelo permite que a organização selecione uma ou mais áreas de processos específicos para aplicar melhorias de qualidade buscando de forma gradativa agregar eficiência e eficácia em seus processos. Como isso, o modelo busca ofertar produtos e serviços criados para satisfazer padrões nacionais e internacionais de qualidade.

Criado pelo *Software Engineering Institute* (SEI), o *Capability Maturity Model Integration* (CMMI) (SEI, 2010) surgiu da necessidade de integrar os diversos modelos de

maturidade (*Capability Maturity Model - CMM*) disponíveis e compatibilizar com a norma ISO/IEC 15504 (ISO/IEC 15504, 2003). As metas para o desenvolvimento do CMMI eram, além de integrar os modelos, eliminar inconsistências, reduzir o custo de implementação do modelo, melhorar o entendimento do modelo e assegurar a consistência com a série ISO/IEC 15504 (GUERRA; COLOMBO, 2009).

O CMMI foi desenvolvido especialmente para software e foi construído a partir do conceito de processo. É composto pelas melhores práticas relativas às atividades de desenvolvimento e manutenção aplicadas a produtos e serviços. Ele abrange práticas que cobrem o ciclo de vida do produto desde a concepção até a entrega e manutenção, e se concentra no trabalho necessário para construção e manutenção do produto em sua totalidade.

Com o modelo CMMI, a organização pode promover a melhoria do processo de software, por meio do aprimoramento da capacidade dos processos ou da maturidade da organização. Este modelo está dividido em duas representações, são elas (SEI, 2010):

1. Representação Contínua: esta representação permite que a organização escolha a prioridade das melhorias de acordo com os objetivos de negócio ou ainda pelas suas áreas de risco. É caracterizado por níveis de capacidade: nível 0 (Incompleto), nível 1 (Executado), nível 2 (Gerenciado), nível 3 (Definido). Os níveis 4 (Gerenciado Quantitativamente) e 5 (Em Otimização) foram removidos dessa última versão do CMMI (versão 1.3).
2. Representação por Estágios: provê uma reconhecida sequência de melhorias, iniciando pelas práticas gerenciais básicas e avançando gradativamente por um caminho predefinido de níveis sucessivos, em que cada nível serve de base para o próximo. Este processo também é caracterizado por níveis de maturidade: nível 1 (Inicial), nível 2 (Gerenciado), nível 3 (Definido), nível 4 (Gerenciado Quantitativamente), nível 5 (Em Otimização).

Tanto os níveis de capacidade como os níveis de maturidade fornecem maneiras para medir quanto as organizações podem melhorar e quanto elas efetivamente melhoram seus processos. Entretanto, a abordagem associada para a melhoria de processo é diferente. A principal contribuição do modelo CMMI e da norma ISO/IEC 15504 é a liberdade que se adquire com a representação contínua, de escolher quais processos devem ser trabalhados naquele momento, respeitando as necessidades prioritárias e particularidades existentes em cada organização (GUERRA; COLOMBO, 2009). No CMM, a representação era apenas por estágios.

O modelo CMMI é composto por 22 áreas de processo, estas juntamente com os seus respectivos níveis de maturidade são apresentados na Tabela 1.

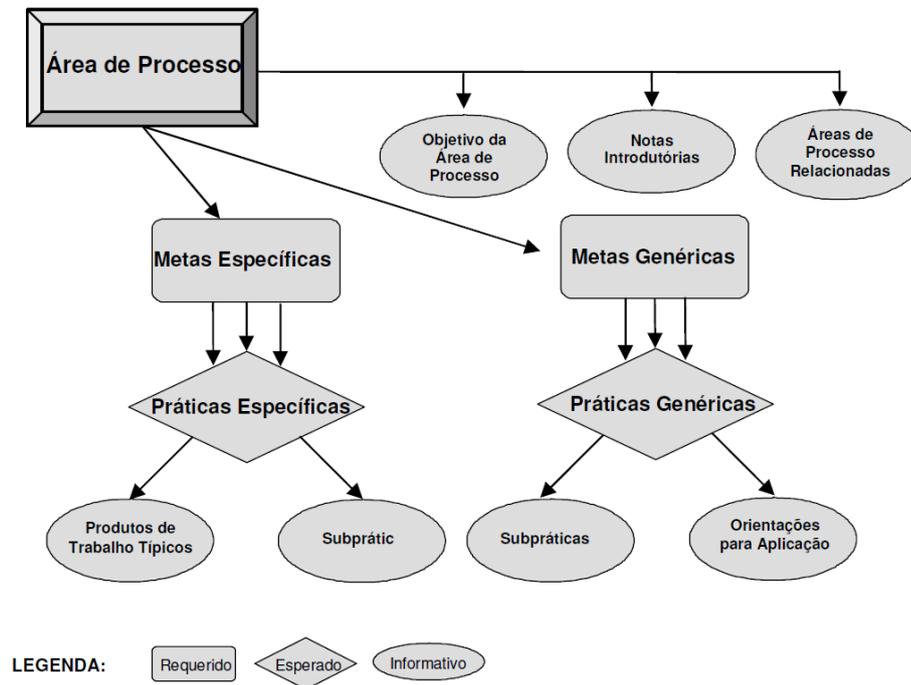
Tabela 1 – CMMI: Áreas de Processo e seus Níveis de Maturidade

Área de Processo	Nível de Maturidade
Análise e Resolução de Causas	5
Gerenciamento de Configuração	2
Análise e Tomada de Decisões	3
Gerenciamento Integrada de Projeto +IPPD	3
Medição e Análise	2
Implantação de Inovações na Organização	5
Definição dos Processos da Organização +IPPD	3
Foco nos Processos da Organização	3
Desempenho dos Processos da Organização	4
Treinamento na Organização	3
Integração de Produto	3
Monitoramento e Controle de Projeto	2
Planejamento de Projeto	2
Garantia da Qualidade de Processo e Produto	2
Gerenciamento Quantitativa de Projeto	4
Desenvolvimento de Requisitos	3
Gerenciamento de Requisitos	2
Gerenciamento de Riscos	3
Gerenciamento de Contrato com Fornecedores	2
Solução Técnica	3
Validação	3
Verificação	3

São ilustrados na Figura 5 os componentes do CMMI, no qual cada nível é constituído por um conjunto de áreas de processos, compostas por metas específicas e/ou genéricas. Cada meta específica é composta por um conjunto de práticas específicas. Uma meta específica descreve as características que devem estar presentes para satisfazer uma área de processo. Uma prática específica é a descrição de uma atividade que é considerada importante para se alcançar a meta específica a ela associada.

Os componentes do modelo são agrupados em três categorias: os requeridos (descrevem o que uma organização deve realizar para implementar uma área de processo), esperados (descrevem o que uma organização pode implementar para satisfazer um componente requerido, orientando os responsáveis por implementar melhorias ou executar avaliações) e informativos (fornecem detalhes às organizações para auxiliá-las na implementação dos componentes requeridos e esperados).

Figura 5 – Componentes do Modelo CMMI



Fonte: (SEI, 2010).

Neste trabalho, o CMMI foi utilizado por ser um modelo de qualidade bastante difundido e utilizado na comunidade de software. O CMMI fornece o conhecimento sobre os processos que são necessários para manter uma organização madura e disciplinada, capaz de prever e melhorar o desempenho da organização e dos seus projetos (JAKOBSEN; JOHNSON, 2008) (ERICSSON et al., 2010). Jing (JING, 2007) ressalta que a aplicação do CMMI ajuda a empresa a modificar e melhorar o processo de trabalho de gestão da inovação e melhorar o nível de controle de processo. Além disso, ajuda a desenvolver produtos com melhor qualidade e eficiência de funcionamento. Ele afirma ainda que não é apenas um grande passo em frente na gestão da inovação, mas também a maturidade da gestão de uma empresa. Outra vantagem é que o CMMI está organizado em uma série de práticas, organizadas em cinco níveis crescentes de maturidade. Cada nível é constituído por um conjunto de áreas de processos, compostas por metas específicas e suas respectivas práticas específicas.

2.4 PROCESSOS DE SOFTWARE

Um processo de software define as práticas e atividades a serem realizadas, as características críticas de cada atividade (como entradas e saídas, os critérios de entrada e

saída e os papéis das partes interessadas), além das relações entre as atividades (XU; RAMESH, 2008). De acordo com Fuggetta (FUGGETTA, 2000) é um conjunto de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos que são necessários para conceber, implantar e manter um produto de software.

Ao focar em processos, as empresas adquirem os fundamentos necessários para enfrentar um mundo em constante mudança, além de maximizar a produtividade das pessoas e o uso da tecnologia, visando maior competitividade. Os processos permitem alinhar a maneira de fazer negócio. Possibilitam explorar a escalabilidade e facilitam a incorporação do conhecimento e das melhores práticas. Processos permitem a otimização de recursos e uma melhor compreensão das tendências de negócio (SEI, 2010). Além disso, acredita-se que uma organização com processos de software bem definidos tem maior probabilidade de desenvolver produtos que atendam às exigências do cliente dentro do cronograma e do orçamento, quando comparada a uma organização com falhas gerenciais e sem processos definidos (SOLINGEN, 2004). O SEI, em seu modelo de qualidade CMMI (SEI, 2010), baseou-se na premissa de gestão de processo de que “a qualidade de um sistema ou produto é altamente influenciada pelo processo utilizado para desenvolvê-lo e mantê-lo”.

2.4.1 Rational Unified Process (RUP)

O *Rational Unified Process (RUP)* (IBM RATIONAL, 2003) é um *framework* de processo de engenharia de software que fornece um conjunto de práticas testadas na indústria para desenvolvimento de software e gerência de projetos (KREBS; SHUJA, 2008). Além de ser um processo, o RUP é tratado como um produto, associado ao *Rational Method Composer (RMC)*, o qual é desenvolvido e mantido pela IBM.

Segundo Bezerra (BEZERRA, 2015), alguns objetivos de um processo de desenvolvimento são: definir quais as atividades a serem executadas ao longo do projeto – quando, como e por quem tais atividades serão executadas – prover pontos de controle para verificar o andamento do desenvolvimento e padronizar a forma de desenvolver software em uma organização.

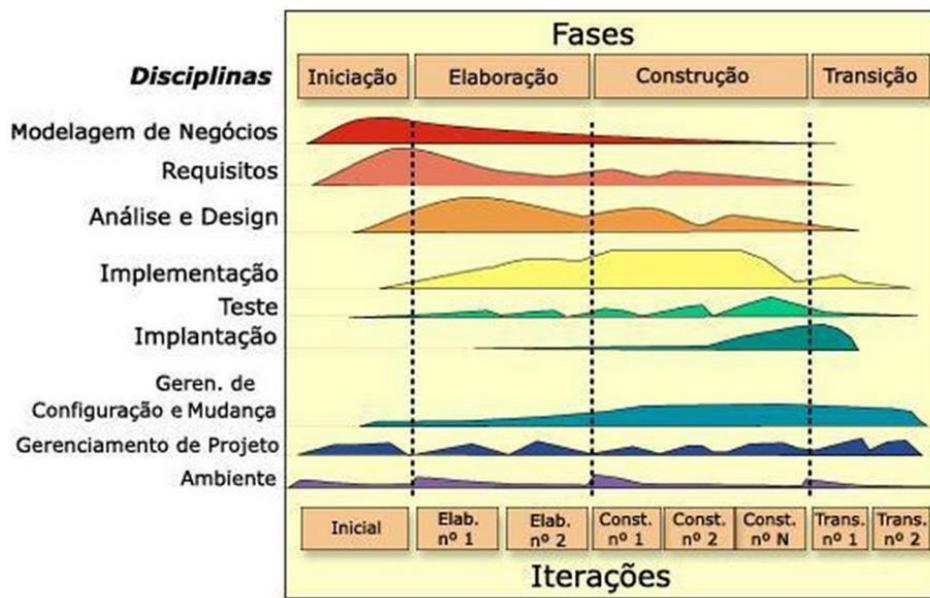
Deste modo, o RUP usa abordagem orientada a objetos e preconiza a utilização da notação *UML (Unified Modeling Language)* (BOOCH; RUMBAUGH; JACOBSON, 2005) para documentação. É organizado em disciplinas (*workflows*), nas quais são distribuídas tarefas, responsabilidades e gerados produtos de trabalho (artefatos). O ciclo de vida de

software do RUP é dividido em quatro fases sequenciais e cada fase pode ser dividida em um número planejado de iterações.

Na Figura 6 pode ser observada que o RUP versão 7.2 organiza o desenvolvimento de software em quatro fases – *Iniciação*, *Elaboração*, *Construção* e *Transição* – nas quais estão inseridas as nove disciplinas, são elas: *Modelagem de Negócios*, *Requisitos*, *Análise e Design*, *Implementação*, *Teste*, *Implantação*, *Gerenciamento de Configuração e Mudança*, *Gerenciamento de Projeto* e *Ambiente*.

Cada fase tem um papel fundamental para que o objetivo seja cumprido, distribuídos entre vários profissionais como o gerente de projetos, analista de sistema, projetista, testador, entre outros. O eixo horizontal representa o tempo, expresso em termos de fases, iterações e marcos. O preenchimento do gráfico representa o volume de cada atividade ao longo das fases e iterações.

Figura 6 – Fases e Disciplinas do RUP



Fonte: (IBM RATIONAL, 2003).

As fases do RUP são descritas a seguir (IBM RATIONAL, 2003):

1. Fase de Concepção/Iniciação: abrange as tarefas de comunicação com o cliente e planejamento. É feito um plano de projeto avaliando os possíveis riscos, as estimativas de custos e prazos, estabelecendo as prioridades, levantamento dos requisitos do sistema e analisando-os preliminarmente.

2. Fase de Elaboração: o objetivo desta fase é criar a *baseline* para a arquitetura do sistema a fim de fornecer uma base estável para o esforço da fase de construção. A arquitetura se desenvolve a partir de um exame dos requisitos mais significativos (aqueles que têm grande impacto na arquitetura do sistema) e de uma avaliação de risco. A estabilidade da arquitetura é avaliada por meio de um ou mais protótipos de arquitetura.
3. Fase de Construção: são desenvolvidos os componentes de software. O principal objetivo desta fase é a construção do sistema de software, com foco no desenvolvimento de componentes e outros recursos do sistema. É na fase de construção que a maior parte de codificação ocorre.
4. Fase de Transição: abrange a fase de testes e a entrega do software ao usuário. O objetivo desta fase é implantar o sistema, tornando-o disponível e compreendido pelo usuário final. As atividades desta fase incluem o treinamento dos usuários finais e também a realização de testes da versão beta do sistema visando garantir que o mesmo possua o nível adequado de qualidade.

Ao mesmo tempo, o RUP é um processo fortemente adaptável às necessidades de cada projeto. Os fundamentos do RUP estão embutidos nas melhores práticas, estas são um conjunto de técnicas para realizar uma determinada tarefa. O RUP engloba muitas das boas práticas do desenvolvimento de software moderno, tornando-se adequado para uma grande quantidade de projetos e organizações que desejam obter valores precisamente quantificados e, ao mesmo tempo, padrões comprovados de sucesso. As seis práticas recomendadas pelo RUP são: desenvolvimento de software iterativo; gerenciamento de requisitos; utilização de uma arquitetura baseada em componentes; modelagem visual do software; análise da qualidade do software e controle das mudanças do software (IBM RATIONAL, 2003).

2.4.2 Adaptação de Processos de Software

Construir um processo a partir do início é arriscado e envolve um alto *overhead*, então os desenvolvedores muitas vezes adaptam-se aos processos e as normas existentes. Vários padrões de processo de software ou modelos de referência, tais como ISO/IEC 12207 (ISO/IEC 12207, 2008) e o RUP (IBM RATIONAL, 2003) fornecem excelentes bases para a adaptação (XU; RAMESH, 2008).

Adaptação de processos de software é o ato de ajustar as definições e/ou particularizar os termos de uma descrição do processo geral para derivar um novo processo aplicável a um

ambiente alternativo e, provavelmente, mais específico. Ou seja, é o ato de adaptar um processo de software padrão para atender às necessidades de uma determinada organização ou projeto (PEDREIRA et al., 2007). Esse desenvolvimento pode envolver a seleção e adaptação de processos existentes (incluindo os processos-padrão), bem como a elaboração de novos processos (SEI, 2010).

Segundo Xu e Ramesh (XU; RAMESH, 2007), adaptação de processo de software envolve: escolher o nível de formalismo do processo, os tipos e conteúdo da documentação produzida e os parâmetros de iterações e *releases*; definir os papéis das partes interessadas e adicionar, remover ou substituir os elementos do processo, como tarefas e artefatos. Os autores salientam que o processo de adaptação é um desafio, visto que o entendimento das características de um projeto afetam as estratégias de adaptação de processos, embora existam os modelos de referência de processos que mostram algumas diretrizes na criação de um processo adaptado (XU; RAMESH, 2008).

Pedreira *et al.* (PEDREIRA et al., 2007) ressaltam a importância do processo adaptado estar em conformidade com as necessidades e com o contexto específico do projeto. Tendo em conta que as organizações são diferentes e que projetos por si só também são diferentes entre si, um processo que pode ser aplicado com sucesso em um projeto, pode falhar em outro. Os autores afirmam ainda que uma adaptação de processos realizada de maneira inadequada pode trazer consequências para a organização, tais como: problemas com orçamento do projeto, tempo de desenvolvimento, qualidade do produto, desenvolvimento de atividades desnecessárias.

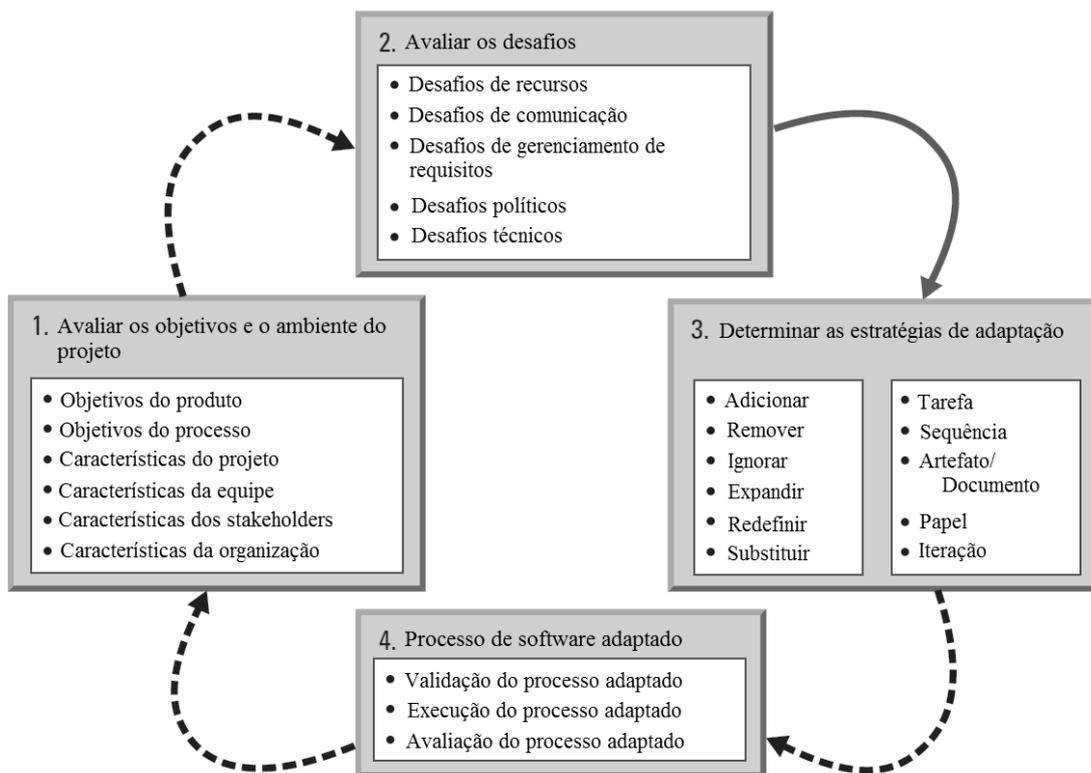
Neste trabalho, os critérios usados para definir as características e o contexto de projetos foram descritos a partir do *Octopus Model*, proposto por Kruchten (KRUCHTEN, 2010). Este modelo destaca oito fatores para contextualizar o projeto, tais fatores influenciam o desenvolvimento de software, portanto devem ser considerados no momento de definir as atividades a serem incorporadas nos processos de software. Os fatores definidos no *Octopus Model* são: tamanho, arquitetura estável, modelo de negócio, distribuição da equipe, taxa de mudanças, idade do sistema, criticidade e controle.

Alguns modelos de maturidade, tais como o CMMI (CHRISSIS; KONRAD; SHRUM, 2006), propõem o uso de guias ou diretrizes de adaptação de processos. Estas diretrizes de adaptação são estabelecidas para guiar os projetos na seleção do modelo de ciclo de vida (dentre as opções aprovadas pela organização) e para adaptar o processo padrão de forma que atenda às especificidades do projeto. Porém, ao contrário do que se poderia esperar, estes modelos não definem efetivamente o que estes guias de adaptação devem conter e fornecem

apenas algumas orientações gerais que não são suficientes para efetivamente ajudar na adaptação (ARMBRUST et al., 2008)(XU; RAMESH, 2008).

Quatro passos principais para a adaptação de processos de software foram propostos por Xu e Ramesh (XU; RAMESH, 2008), conforme mostrado na Figura 7. O objetivo do trabalho era ajudar os gerentes de projeto a determinar como adaptar um processo de software para enfrentar desafios específicos de um projeto. Os passos criados foram: avaliar os objetivos e o ambiente do projeto; avaliar os desafios que o projeto precisa enfrentar; determinar as estratégias do processo de adaptação que possam mitigar esses desafios e; validar, executar e avaliar o processo adaptado.

Figura 7 – Passos para Adaptação de Processos de Software



Fonte: (XU; RAMESH, 2008).

2.4.3 Linhas de Processos de Software

Abordagens baseadas em reuso do conhecimento relacionado a processos de software são largamente utilizadas em adaptação de processos, tais como componentes, arquiteturas e Linhas de Produtos de Software (LPS) (WASHIZAKI, 2006a). A adaptação de processos pode ser feita partindo de um processo padrão da organização ou por meio de estruturas de

reutilização de processos, como componentes de processos (CHRISISS; KONRAD; SHRUM, 2006).

Da mesma forma, os modelos de maturidade (CHRISISS; KONRAD; SHRUM, 2006)(SOFTEX, 2011) também estabelecem que os processos podem ser definidos com base em unidades menores e reutilizáveis de processos. Assim, os processos são decompostos em componentes de processo e podem ser conectados para instanciar processos novos e mais complexos (MAGDALENO et al., 2007). O fato de estar reutilizando processos permite fazer uso do conhecimento e da experiência adquiridos previamente. Desta forma, é possível agilizar a tarefa de adaptação de processos.

A ideia de Linha de Processos de Software (LPrS) (ALEIXO et al., 2010)(BARRETO; MURTA; ROCHA, 2011)(JAUFMAN; MÜNCH, 2005)(ROMBACH, 2006)(TERNITÉ, 2009)(WASHIZAKI, 2006a) surge da transferência do paradigma de Linhas de Produtos (NORTHROP, 2002) para o domínio de processos, em outras palavras, linhas de processos são linhas de produtos cujos produtos são processos de software. Dessa maneira, uma LPrS funciona como uma fábrica, que instancia produtos parecidos, cada um com um conjunto de características, por meio da composição de componentes existentes. Um componente é considerado algo relevante para ser reutilizado em outras definições de processo e encapsula uma série de informações de processo, tais como: atividades, produtos de trabalho requeridos e produzidos, responsáveis pela execução, entre outros (NUNES *et al.*, 2010).

A adoção de linhas de processos permite alavancar a reutilização de componentes de processo individuais para arquiteturas de processo completas, contendo diversos componentes inter-relacionados (ROMBACH, 2006)(WASHIZAKI, 2006a)(BARRETO; MURTA; ROCHA, 2011). Vale ressaltar que arquitetura de processos de software é definida como “uma estrutura de processos que reflete semelhanças e variabilidades em um conjunto de processos que compõem uma Linha de Processos a partir da perspectiva de otimização global” (WASHIZAKI, 2006b).

De acordo com Washizaki (WASHIZAKI, 2006a), uma LPrS pode ser definida como um conjunto de processos de um determinado domínio de problema ou com um determinado propósito, que tem características em comum e é construído baseado em ativos reutilizáveis de processos. Autores como Barreto *et al.* (BARRETO; MURTA; ROCHA, 2011) e Magdaleno (MAGDALENO, 2010) propõem o uso de LPrS como forma de possibilitar a reutilização de componentes e contextualização do processo.

Neste sentido, os objetivos de uso de linhas de processos são: i) aumentar a produtividade da atividade de adaptação de processos, diminuindo o esforço necessário para

realizá-la; ii) aumentar a qualidade e adequação dos processos gerados (reutilização do conhecimento de especialistas e de dados sobre utilização); iii) representar variabilidades e semelhanças entre processos para potencializar a reutilização; e iv) diminuir os riscos de uma adaptação inadequada do processo (ROMBACH, 2006)(JAUFMAN; MÜNCH, 2005)(PEDREIRA et al., 2007).

3 CONTEXTO DA PESQUISA

Tanto a abordagem de avaliação de artefatos de software, quanto a adaptação de processos de software com LPrS já foram discutidos em trabalhos passados desenvolvidos no Grupo de Pesquisa em Engenharia de Software (PEEnSo) no qual essa dissertação está inserida.

A principal contribuição do trabalho de Bertuol (BERTUOL, 2014) foi a proposta de um mecanismo de avaliação, capaz de apoiar equipes de desenvolvimento de software – especialmente as menos experientes – na definição de propósitos de qualidade almejados para o produto final por meio da avaliação de artefatos de software.

A adaptação de processos foi o tema do trabalho de Lorenz (LORENZ, 2014). Este propõe uma abordagem sistemática para adaptação de processos de software a partir de LPrS e informações sobre as características dos projetos. A abordagem proposta utiliza o reúso de elementos de processos, previamente definidos, gerando processos planejados e ágeis. A LPrS adaptada é elaborada a partir da recuperação de elementos previamente cadastrados nas arquiteturas de processos, ou seja, são recuperadas atividades contextualizadas de acordo com os requisitos de adaptação e o contexto do projeto.

Um dos objetivos deste trabalho foi integrar as pesquisas desenvolvidas por Bertuol (BERTUOL, 2014) e Lorenz (LORENZ, 2014). Foi utilizada para esta dissertação a adaptação de processos conforme Lorenz (LORENZ, 2014) e foi realizada uma expansão da pesquisa desenvolvida por Bertuol (BERTUOL, 2014), no qual o processo de elaboração do plano de avaliação dos artefatos tornou-se uma abordagem semi-automatizada. Além disso, foi adicionado um requisito de adaptação visando a qualidade. Os critérios disponibilizados na abordagem são: riscos associados a um projeto (FONTOURA, 2006) (LORENZ, 2014), requisitos de segurança (PERSCH, 2014) e agora, práticas do modelo de avaliação de processos CMMI proposta neste trabalho.

3.1 UMA ABORDAGEM PARA AVALIAÇÃO DA QUALIDADE DE ARTEFATOS DE SOFTWARE

Bertuol (BERTUOL, 2014) propôs uma abordagem sistemática para a definição de um *framework* de qualidade, cujo propósito é fornecer uma estrutura linear aos avaliadores, engenheiros e demais interessados, na validação de artefatos de software gerados ao longo de um processo de desenvolvimento de software. O *framework* é composto por um metamodelo

que apresenta um conjunto de elementos considerados pertinentes para a avaliação da qualidade de artefatos de software, além de incluir um processo de avaliação e uma ferramenta de apoio.

O metamodelo tem o objetivo de ajudar os interessados na adoção de uma visão comum acerca dos requisitos de qualidade pretendidos para um projeto específico, ao mesmo tempo em que permite uma decomposição estruturada dos elementos, conceitos e relacionamentos necessários a essa visão (BRONDANI; BERTUOL; FONTOURA, 2015). A definição do metamodelo baseou-se em três requisitos básicos, propostos por Trendowicz e Punter (TRENDOWICZ; PUNTER, 2003), que são: flexibilidade, reusabilidade e transparência.

A flexibilidade está associada à dependência do contexto de qualidade do software. O modelo de avaliação deve ser suficientemente flexível ao ponto de adequar-se às diferentes abordagens que envolvem o ambiente organizacional de cada projeto de software. No contexto deste trabalho, a flexibilidade também está associada às diferenças existentes entre os próprios artefatos que são produzidos durante as fases do processo de desenvolvimento do software. Documentação, modelos UML, código-fonte, entre outros artefatos, possuem características próprias e, assim, o *framework* de avaliação deve permitir, aos avaliadores, identificarem essas características e definirem quais as melhores propostas de avaliação para cada uma (BERTUOL, 2014).

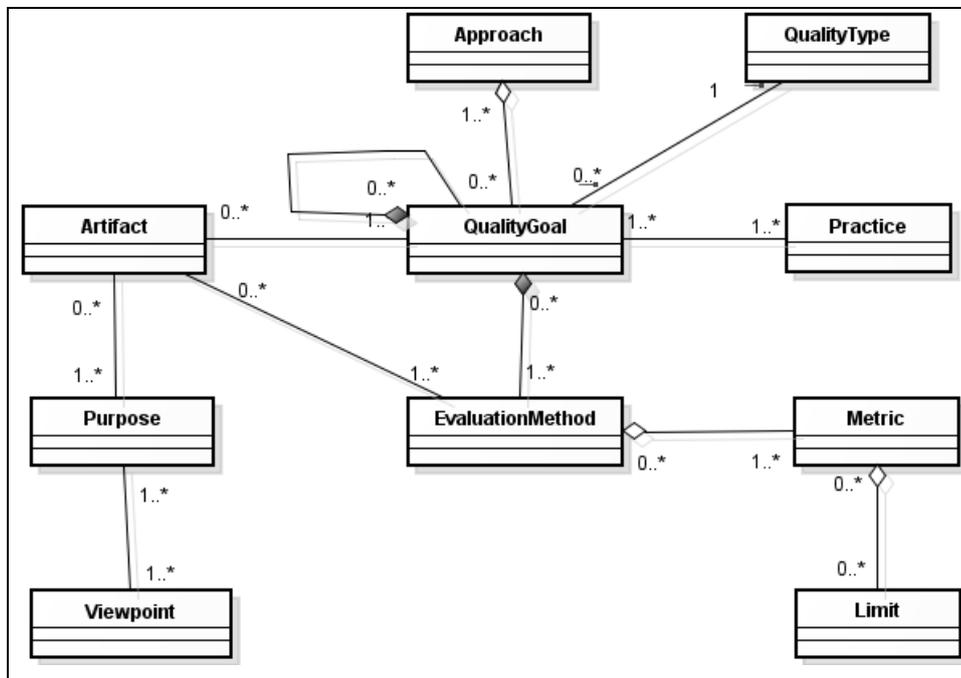
Paralelamente, a reusabilidade está associada a necessidade de se reter o conhecimento com experiências passadas e usá-lo em projetos futuros, o que impacta na necessidade de se desenvolver projetos no menor tempo e custo possível e, conseqüentemente, aumentar a rentabilidade desses projetos. Obviamente que a reutilização em um processo de avaliação depende diretamente da similaridade entre os projetos, porém, o reúso pode considerar tanto os dados de medições, quanto às características de qualidade e seus relacionamentos. Outra vantagem é que à medida que o modelo vai sendo reusado, é possível ajustá-lo, tornando-o mais preciso e eficiente (TRENDOWICZ; PUNTER, 2003).

Por fim, um modelo de avaliação deve fornecer uma análise racional e transparente de como as características e subcaracterísticas de qualidade se relacionam entre si e qual o impacto de uma sobre a outra. Por exemplo, a equipe de desenvolvimento deve perceber que a modularização de um diagrama de classes – que posteriormente servirá para definir a estrutura das tabelas de um banco de dados – permite um melhor entendimento da estrutura do software. Contudo, a normalização excessiva dessas tabelas pode, reconhecidamente, afetar o desempenho do banco de dados e do sistema como um todo. Uma solução encontrada para

contornar o problema da transparência é permitir que as próprias partes interessadas definam, baseados em consenso, as métricas e métodos de avaliação mais relevantes e que melhor representem cada artefato e, ao mesmo tempo, usem o metamodelo para solucionar possíveis ambiguidades ou redundâncias (TRENDOWICZ; PUNTER, 2003).

A estrutura do metamodelo, ilustrada na Figura 8, bem como suas relações, foi elaborada por Bertuol (BERTUOL, 2014) a partir dos modelos de qualidade e visa organizar o conhecimento dos avaliadores na busca por métricas e métodos de avaliação que melhor representem os seus propósitos de qualidade.

Figura 8 – Metamodelo de Qualidade



Fonte: (BERTUOL, 2014).

Seus principais constructos são explicados a seguir:

- Abordagem (*Approach*): é especificada para representar instâncias de diferentes metodologias ou paradigmas que podem ser usados no desenvolvimento de softwares.
- Artefato (*Artifact*): são quaisquer produtos de saída de uma atividade, como por exemplo, código fonte, modelos UML, documentos, especificações, e assim por diante. Simultaneamente, as atividades são executadas por papéis que representam pessoas e/ou ferramentas.
- Metas de Qualidade, Tipo de Qualidade (*QualityGoal*, *QualityType*): as metas de qualidade são a definição clara e compreensível de quais atributos ou características de

qualidade um determinado *stakeholder* está interessado para um determinado artefato de software. Essas metas são específicas para cada artefato, por exemplo, um Modelo de Caso de Uso tem como meta a compreensão dos requisitos, ao passo que um código fonte deve ser completo e consistente. Para melhor estruturar a adaptação do processo de avaliação e organizar os artefatos com base em padrões organizacionais, o metamodelo permite que as metas de qualidade possam ser identificadas por meio de um *QualityType*. Isso ajuda a arquetá-las em níveis de abstração diferentes, que podem ser criados pela equipe de avaliação ou baseados em algum dos modelos existentes.

- **Propósito (*Purpose*):** descrevem qual o propósito de uma parte interessada em atingir uma determinada meta de qualidade. Por exemplo, um usuário final pode estar interessado em facilidade de uso como meta, ao mesmo tempo em que um desenvolvedor busca completude e correção.
- **Ponto de Vista (*Viewpoint*):** o ponto de vista é usado para indicar as partes interessadas envolvidas nas metas de qualidade pretendidas. É importante notar a diferença entre as partes interessadas e papéis, definidos na classe *Purpose*. Embora possa haver uma sobreposição de definições, há uma distinção no propósito de cada uma. Os primeiros estão diretamente interessados na busca pela meta de qualidade, enquanto os últimos executam as atividades e se responsabilizam pelos artefatos produzidos.
- **Prática (*Practice*):** são os meios pelos quais se pode alcançar uma determinada meta de qualidade. Por exemplo, o uso de convenções para modelagem é uma prática que pode levar ao desenvolvimento de modelos UML mais corretos, consistentes e compreensíveis; refatoração do código fonte é uma prática para melhorar a reusabilidade; e assim sucessivamente.
- **Métodos de Avaliação, Métrica, Limite (*EvaluationMethod, Metric, Limit*):** os métodos de avaliação identificam a forma como uma determinada meta de qualidade deve ser avaliada. Os métodos de avaliação podem ser quantitativos ou qualitativos. Esses métodos são generalizados para suportar tanto métodos específicos (por exemplo, simulações, inspeções, *checklists*, etc.) quanto métricas específicas, como por exemplo, NUC (*number of classes per use case*) ou NCU (*number of use cases per class*) para modelos UML ou KLOC (número de erros por mil linhas de código). Para isso, cada métrica é definida com base em uma unidade e nos valores limite máximo, mínimo e um valor tido como aceitável (classe *Limit*). É importante destacar que há casos em que os valores dependem do contexto da avaliação. Dessa forma, tendo em

vista que cada métrica possui particularidades, a definição de valores e unidades, embora recomendada, é opcional e que os valores atribuídos na inserção de uma determinada métrica servirão, apenas, como referência para os avaliadores, que poderão alterá-los no momento em que estiverem definindo o plano de qualidade, durante o processo de avaliação.

3.2 UMA SISTEMÁTICA BASEADA NO REÚSO DE ATIVIDADES PARA ADAPTAÇÃO DE PROCESSOS

Este projeto iniciou-se com o trabalho de Lorenz *et al.* (LORENZ et al., 2014), cujo principal objetivo foi definir uma abordagem sistemática para a adaptação de processos de software a partir de LPrS e informações sobre as características dos projetos. A abordagem permite a reutilização de elementos do processo, previamente definidos, permitindo gerar processos ágeis e planejados. A fim de validar o trabalho, foi desenvolvida uma ferramenta *Web* denominada *Metamodel for Tailoring Process tool (MfTPt)* para apoiar o processo de adaptação, melhorando a técnica de seleção de componentes de processos reutilizáveis.

A ferramenta MfTPt foi organizada em dois módulos, que são: o módulo principal e o módulo para adaptação de processos de software. O módulo principal contém todas as funcionalidades para a criação do repositório de componentes de processos, tais como: cadastro dos artefatos, tarefas, papéis, atividades, requisitos de adaptação, atributos para contextualização de atividades de projetos e definição de arquiteturas de processo. O módulo de adaptação de processos de software, por sua vez, é sistematizado por meio de quatro etapas (Figura 9), que são descritas a seguir.

Figura 9 – Sistemática de Adaptação de Processos de Software



Fonte: (LORENZ et al., 2014).

1. *Definição das características do projeto*: os critérios usados para contextualização de projetos foram descritos a partir do *Octopus Model* (KRUCHTEN, 2010). O

objetivo da contextualização é definir se componentes de processos ágeis ou planejados são mais indicados ao projeto em questão. O contexto do projeto é comparado ao contexto de cada componente de processo armazenado no repositório, visando identificar componentes de processo com contexto mais similar ao do projeto e que satisfazem um determinado critério. É importante que o processo adaptado esteja em conformidade com as necessidades e ao contexto específico do projeto. Brasil *et al.* (BRASIL; FONTOURA; ALVARO, 2013) também explora a prática de adaptação de processos com base em uma situação específica de um projeto, esta situação é indicada a partir de um contexto previamente definido, ora por características do projeto/organização, ora por metas ou resultados a serem alcançados.

2. *Seleção dos requisitos de adaptação e da arquitetura de processos*: processos de desenvolvimento possuem necessidades específicas que devem ser contempladas. Esta etapa visa, independentemente do tipo de necessidade específica, selecionar componentes de processo que venham a satisfazer essas necessidades e incorporá-los ao processo. Para cada componente definido na arquitetura são recuperadas atividades que possuem finalidades semelhantes. Pode-se definir uma arquitetura a partir de abordagens ágeis, abordagens planejadas, ou ainda combinar as características de métodos ágeis e planejados em um único processo de desenvolvimento.
3. *Priorização das atividades*: uma das questões críticas das abordagens de adaptação de processos de software está relacionada com a escolha das atividades mais apropriadas de acordo com as características do projeto. Os componentes previamente definidos para a arquitetura são recuperados de acordo com os requisitos de adaptação para criar a LPrS. Com isso, os componentes são priorizados por meio de um algoritmo baseado na técnica *Analytic Hierarchy Process* (AHP). Além dessa técnica, outros dois métodos de priorização foram implementados por Persch (PERSCH, 2014), são eles: Tomada de Decisão Interativa Multicritério (TODIM) e *Technique for Order Preference by Smilarity the Ideal Solution* (TOPSIS) que, a partir das suas teorias e diretrizes de cálculo, apresentam as melhores alternativas dos componentes baseando-se no contexto do projeto que é apresentado. As alternativas para priorização correspondem às atividades armazenadas no repositório de componentes e o objetivo é priorizá-las adequadamente para satisfazer os requisitos de adaptação. Detalhes sobre a técnica

de priorização de componentes pode ser obtida em (PEREIRA; SEVERO; FONTOURA, 2012).

4. *Criação da linha de processo de software adaptada:* a partir da arquitetura de processos definida e das atividades recuperadas, priorizadas e selecionadas com os métodos matemáticos, tem-se a elaboração da LPrS adaptada de acordo com as características situacionais do projeto.

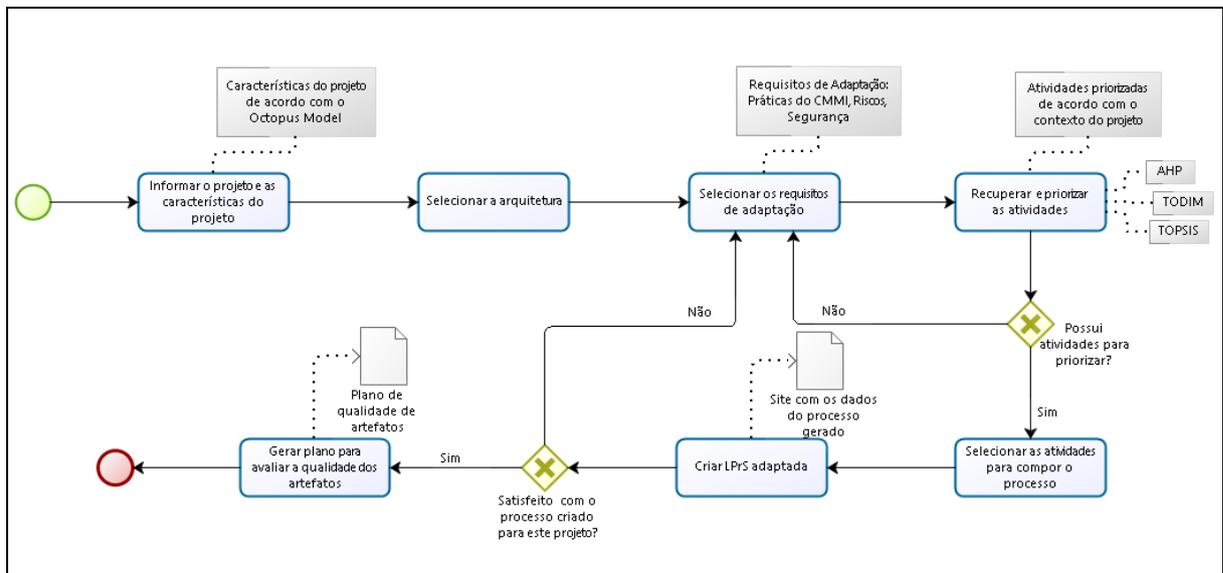
Como resultado do trabalho de Lorenz *et al.* (LORENZ et al., 2014) tem-se que essa abordagem ajuda o engenheiro de processos a tomar decisões para a seleção de um conjunto de componentes de processo adequados às exigências de adaptação e ao contexto do projeto para serem instanciados nos pontos de variação descritos pela linha de processo.

4 FRAMEWORK PARA AVALIAÇÃO DA QUALIDADE DE ARTEFATOS

Este capítulo descreve, como parte do *Quality Assessment of Artifacts in Process – Framework* (QAPro-F) proposto, um metamodelo, criado com o objetivo de apoiar a descrição que compõe a abordagem proposta nesta dissertação, além da descrição da base de conhecimento suportado pelo CMMI, um processo de avaliação para artefatos de software da linha adaptada e uma ferramenta de apoio.

Para estruturar as principais atividades e a sequência de ações que descrevem a abordagem, na Figura 10 é exibida a sistemática proposta para adaptação de processos de software, juntamente com a avaliação da qualidade dos artefatos, usando a notação *Business Process Modeling Notation* (BPMN). A modelagem da abordagem é composta pelas seguintes tarefas: i) informar o projeto e as características do projeto; ii) selecionar a arquitetura; iii) selecionar os requisitos de adaptação; iv) recuperar e priorizar as atividades; v) selecionar as atividades para compor o processo; vi) criar LPrS adaptada; e vii) gerar plano para avaliar a qualidade dos artefatos.

Figura 10 – Modelagem da Abordagem de Adaptação de Processos de Software com a Avaliação da Qualidade dos Artefatos



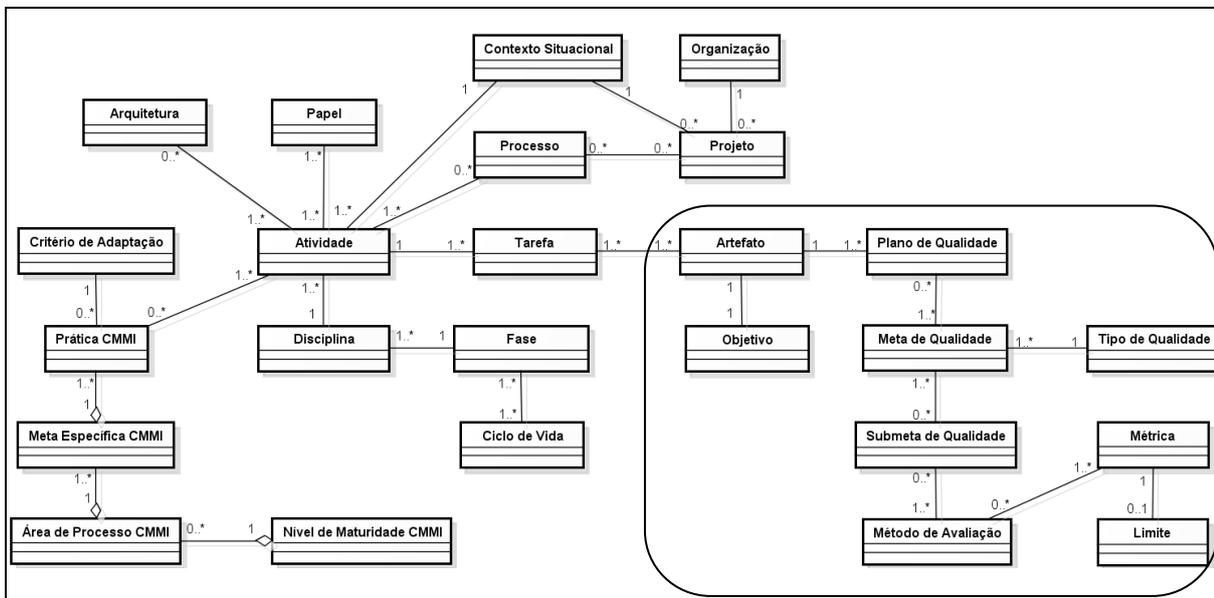
4.1 METAMODELO

Metamodelos são comumente utilizados na engenharia de software orientada a modelos. De acordo com Lee *et al.* (LEE; SHIM; WU, 2002), um metamodelo serve para

criar um outro modelo conceitual, ou seja, é uma forma de descrever como um modelo deve ser criado. Eles definem a estrutura, a semântica e as restrições para uma família de modelos, isto é, grupos de modelos que compartilham uma sintaxe e uma semântica comuns (MELLOR et al., 2004).

O metamodelo denominado *Quality Assessment of Artifacts in Process - Metamodel* (QAPro-M), ilustrado na Figura 11, é uma evolução do metamodelo proposto por (BERTUOL, 2014). A necessidade de evoluir o metamodelo surgiu da carência de possuir um metamodelo que forneça tanto elementos de processos quanto elementos de qualidade de software. O QAPro-M tem como escopo um conjunto de conceitos e relacionamentos entre os elementos de processos e de qualidade que visam mapear todos os conceitos do domínio da abordagem de avaliação da qualidade de artefatos em uma LPrS adaptada. Sendo assim, o metamodelo criada objetiva estabelecer o vocabulário comum que é essencial para que as pessoas compreendam umas às outras. Portanto, é importante estabelecer relações semânticas entre os conceitos utilizados na definição dos processos, assim como na avaliação de artefatos. Desta forma, o metamodelo funciona como uma fonte unificada de referência, a partir do qual é gerado um repositório de conhecimento compartilhado.

Figura 11 – *Quality Assessment of Artifacts in Process - Metamodel (QAPro-M)*



O metamodelo criado é formado por 25 classes que são detalhadas a seguir. A classe central do metamodelo proposto é a *atividade*, um conjunto de atividades de uma mesma área formam uma *disciplina*. Deste modo, essas disciplinas são distribuídas por meio de *fases*

iterativas, estas correspondem aos intervalos de tempo entre os marcos do projeto, o RUP classifica 4 fases: Concepção, Elaboração, Construção e Transição. Em vista disso, essas fases formam o *ciclo de vida* do processo de desenvolvimento.

Por sua vez, as atividades são constituídas de *tarefas* e cada atividade possui um ou mais *papéis*. Papel representa os cargos executados por indivíduos em um projeto, tais como: analista de sistemas, gerente de projeto, gerente de configuração, usuário, entre outros. *Arquiteturas* são formadas por um conjunto de atividades e uma atividade pode pertencer a mais de uma arquitetura. A arquitetura define a estrutura principal que o processo deve conter e determina os seus componentes principais e como é o relacionamento entre eles.

A classe *projeto* é responsável por representar os projetos definidos para as *organizações*. Desta forma, cada organização pode possuir um ou mais projetos e cada projeto pode ter um ou mais *processos*, permitindo assim, que se um processo após sua criação não tenha atendido às expectativas da organização, este possa ser evoluído criando uma nova versão a partir do processo atual. No momento da criação de um projeto é possível definir um *contexto situacional* para ele. Para contextualizar um projeto, utilizamos o *Octopus Model* (KRUCHTEN, 2010), com os seus oito fatores.

A adaptação de processos considera a caracterização das atividades e requisitos de adaptação. Deste modo, cada atividade possui um ou mais *critérios de adaptação*. Estes tem a função de definir os requisitos que podem ser satisfeitos pelo processo da instanciação de uma determinada atividade, e assim, recuperar do repositório as atividades que satisfaçam os critérios de adaptação definidos pelo engenheiro de processos ou pelo responsável em adaptar ou definir o processo da organização.

Em relação as classes do CMMI (descrito na Seção 2.3) cada *nível de maturidade* é constituído por um conjunto de *áreas de processos*, e estas são compostas por *metas específicas*. Cada meta específica é composta por um conjunto de *práticas* específicas.

Paralelamente aos elementos já citados, o metamodelo é composto por elementos de qualidade – demarcados na Figura 11 – estes são considerados pertinentes para a avaliação da qualidade de artefatos de software e estão presentes nos planos de qualidade dos artefatos. Os elementos de qualidade destacado na Figura 11 já foram explicados separadamente na Seção 3.1. No entanto, vale ressaltar que foram realizados pequenos ajustes em relação ao metamodelo da Figura 8, as alterações foram feitas mais especificamente nos relacionamentos entre os elementos.

A classe *artefato* representa os produtos de trabalho gerados mediante a execução das tarefas. São considerados exemplos de artefatos os protótipos de interface do usuário,

modelos de casos de uso, contratos, *releases* de software, relatórios, entre outros. Os artefatos estão associados a um *objetivo*, o qual visa identificar o propósito que descreve a intenção de um artefato dentro do ciclo de vida do processo de software, juntamente com o motivo pelo qual ele deve ser avaliado.

Por sua vez, a classe *plano de qualidade* diz respeito ao conjunto de elementos responsáveis pela avaliação dos artefatos, os planos estão relacionados a classe *meta de qualidade*. Esta representa quais atributos ou características de qualidade um determinado *stakeholder* está interessado para um determinado artefato de software. O metamodelo permite que as metas de qualidade possam ser identificadas por meio de um *tipo de qualidade*. Um exemplo de tipo de qualidade é a ISO/IEC 9126, que possui 6 características e estas são subdivididas em *submetas de qualidade*.

No entanto, o *método de avaliação* identifica a forma como uma determinada meta de qualidade deve ser avaliada. Esses métodos são generalizados para suportar tanto métodos específicos quanto uma *métrica* específica. Para isso, cada métrica é definida com base no valor *limite*, ou seja, um valor tido como aceitável.

Os elementos de qualidade, instanciados a partir do metamodelo proposto, são armazenados em um repositório, que consiste em um conjunto de planos de avaliação da qualidade de artefatos. Este repositório deve evoluir com o tempo e com a maturidade da organização.

4.2 BASE DE CONHECIMENTO CONSTRUÍDA A PARTIR DO MODELO DE QUALIDADE CMMI

Nesta seção é descrito como o modelo de qualidade CMMI foi utilizado como requisito de adaptação, além de apresentar a elaboração tanto da arquitetura de processo quanto da LPrS para as disciplinas de Requisitos e Configuração e Gerenciamento de Mudanças do RUP.

4.2.1 Uso do CMMI como Requisito de Adaptação

A abordagem de adaptação de processos já possui dois requisitos de adaptação, que são derivados de riscos (LORENZ, 2014) e segurança (PERSCH, 2014). Além disso, visando tornar os requisitos de adaptação mais completos, uma das propostas desta dissertação foi adicionar um requisito de qualidade. Desta forma, foi escolhida as práticas do modelo de

qualidade mais difundido do mundo (JING, 2007), o CMMI (SEI, 2010), descrito na Seção 2.3.

Dentre as 22 áreas de processo do CMMI, as áreas de Gerenciamento de Requisitos e Gerenciamento de Configuração foram o foco desta dissertação. Estas foram escolhidas dada a importância das mesmas, visto que um projeto que não possui os requisitos bem definidos e gerenciados há uma maior probabilidade de não conseguir atender seus objetivos. Portanto, assegurar que o gerenciamento de requisitos seja empregado é uma fundamental competência para o sucesso dos projetos. Do mesmo modo, a gerência de configuração é essencial para produzir software de qualidade, pois mudanças durante o desenvolvimento de software são inevitáveis. Contudo, independentemente do tipo de necessidade específica, a ideia é selecionar elementos de processo que venham a satisfazer essas necessidades específicas e incorporá-los ao processo.

A ideia deste trabalho é que o usuário da ferramenta de adaptação de processos possa escolher o nível de maturidade organizacional que deseja alcançar, assim como a área de processo, a meta específica e finalmente selecionar quais práticas do CMMI deseja satisfazer. Então, as atividades que satisfazem essas práticas serão buscadas no repositório a fim de formar o processo adaptado de acordo com a meta que deseja alcançar de uma determinada área de processo do CMMI.

Vale salientar que o CMMI especifica “*o que deve*” ser feito no processo de software, mas não especifica “*como*”. O RUP descreve um *framework* de processo de desenvolvimento de software, especificando como desenvolver software, isto é, as atividades que devem ser seguidas, os artefatos que devem ser gerados, os responsáveis e o relacionamento entre as atividades. Portanto, neste trabalho, foram selecionados os elementos do RUP das disciplinas de Requisitos e Configuração e Gerenciamento de Mudanças, e estes foram associados às práticas do CMMI que eles podem apoiar a fim de dar suporte às organizações que objetivam alcançar níveis de maturidade do CMMI.

A Tabela 2 mostra um exemplo de associação entre a prática do CMMI *Obtain Commitment to Requirements*, da meta específica *Manage Requirements*, da área de processo Gerenciamento de Requisitos, nível 2 do CMMI, com os artefatos e atividades do RUP. Vale ressaltar que as associações feitas são sugestões propostas por este trabalho. Artefatos e atividades marcados com (R) são correspondentes a disciplina de Requisitos e marcadas com (C) são da disciplina de Configuração e Gerenciamento de Mudanças do RUP.

Tabela 2 – Associação entre Práticas do CMMI com as Atividades e Artefatos do RUP

CMMI: Level 2 - Process Area: Requirements Management (REQM)			
Specific Goals CMMI	Specific Practices CMMI	Artifacts RUP	Activities RUP
SG 1 Manage Requirements	SP 1.2 Obtain Commitment to Requirements	Requirements Management Plan (R) Stakeholder Requests (R) Change Request (C)	Analyze the Problem (R) Understand Stakeholder Needs (R) Define the System (R) Manage the Scope of the System (R) Refine the System Definition (R) Manage Changing Requirements (R) Manage Change Requests (C)

Por exemplo, para a prática “*Obtain Commitment to Requirements*” foram analisadas suas respectivas subpráticas e verificado quais artefatos estavam relacionados. Portanto constatou-se que na disciplina de Requisitos do RUP, os artefatos relacionados a essa prática do CMMI foram o *Requirements Management Plan* e o *Stakeholder Requests*. Na disciplina de Configuração e Gerenciamento de Mudanças do RUP foi o artefato *Change Request*. Após essa análise, as atividades correspondentes a esses artefatos são recuperadas, neste caso foram as atividades: *Analyze the Problem*, *Understand Stakeholder Needs*, *Define the System*, *Manage the Scope of the System*, *Refine the System Definition*, *Manage Changing Requirements* da disciplina de Requisitos. Além dessas, na disciplina de Configuração e Gerenciamento de Mudanças do RUP foi encontrada a atividade *Manage Change Requests*.

As demais associações são encontradas no Apêndice A e, como já mencionado, foi dado ênfase para as práticas das áreas de processo de Gerenciamento de Requisitos e Gerenciamento de Configuração do CMMI.

4.2.2 Elaboração da Arquitetura de Processo baseada no RUP

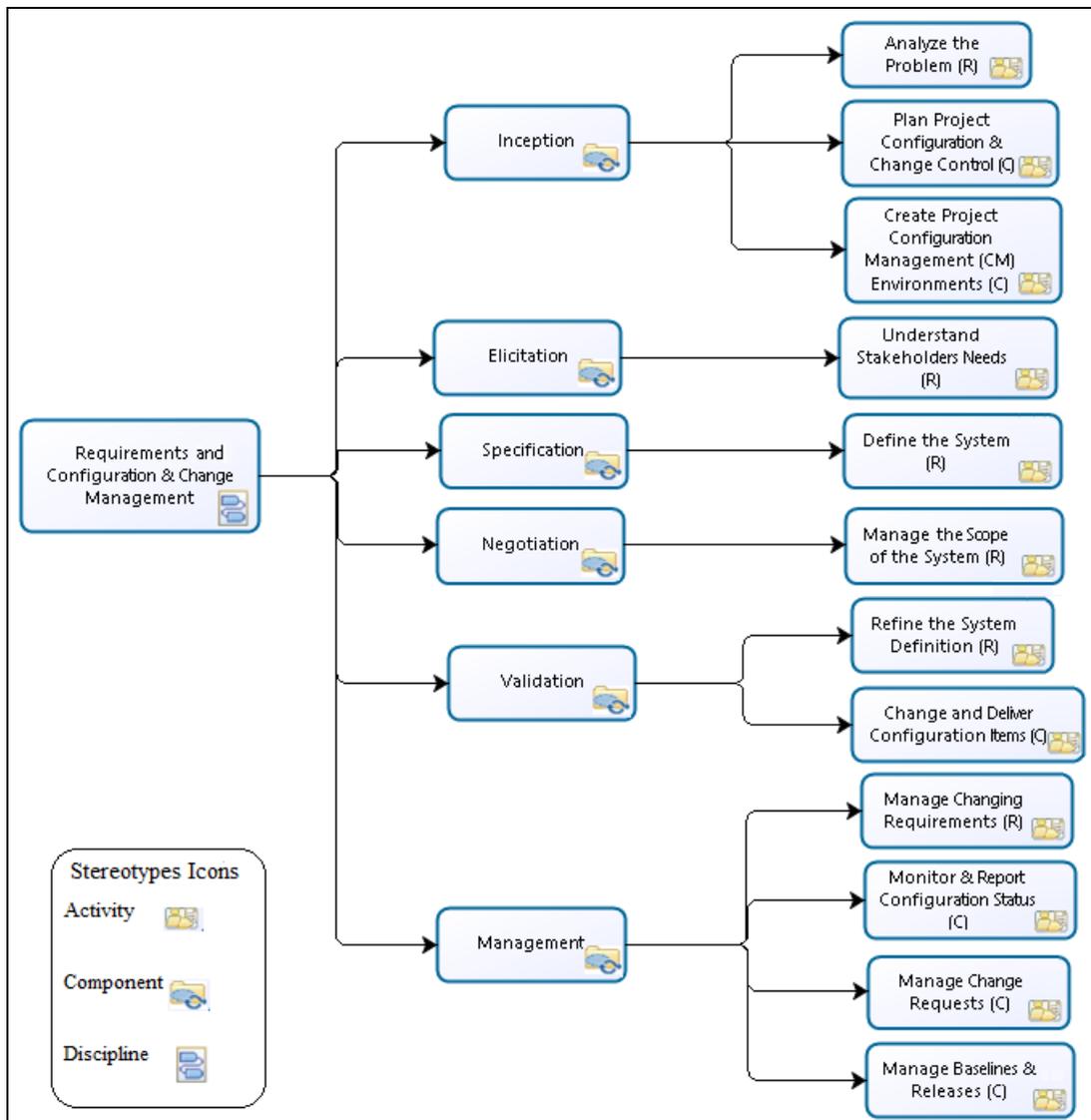
Arquiteturas de processo são compostas por elementos utilizados para definir processo de software, que são instanciados e armazenados em um repositório. O repositório utilizado é formado por elementos descritos nas metodologias: RUP (IBM RATIONAL, 2003), SCRUM (SCHWABER; BEEDLE, 2001), *Extreme Programming* (XP) (BECK, 2004); e padrões de processo e organizacionais (COPLIEN; ALEXANDER, 1996). Esses padrões descrevem soluções para problemas recorrentes e podem ser utilizados na construção e adaptação de processo de software.

Vale ressaltar que a arquitetura de processo define a estrutura principal que o processo deve conter, determina os principais componentes e como é o relacionamento entre eles, não necessariamente definindo os detalhes destes componentes (BARRETO; MURTA; ROCHA,

2011). Posteriormente, uma LPrS adaptada é elaborada por meio da recuperação dos elementos previamente cadastrados nas arquiteturas de processos, ou seja, a utilização da arquitetura proporciona a recuperação de um conjunto de atividades que são priorizadas de acordo com o contexto do projeto e os requisitos de adaptação. Desta maneira, facilita a adequação do processo utilizado para diferentes contextos de projetos.

Em vista disso, para este trabalho, foi definida uma arquitetura para as disciplinas de Requisitos e Configuração e Gerenciamento de Mudanças do RUP (IBM RATIONAL, 2003). Esta arquitetura, ilustrada na Figura 12, é formada por um conjunto de atividades do RUP.

Figura 12 – Arquitetura definida para a disciplina de Requisitos e Configuração e Gerenciamento de Mudanças

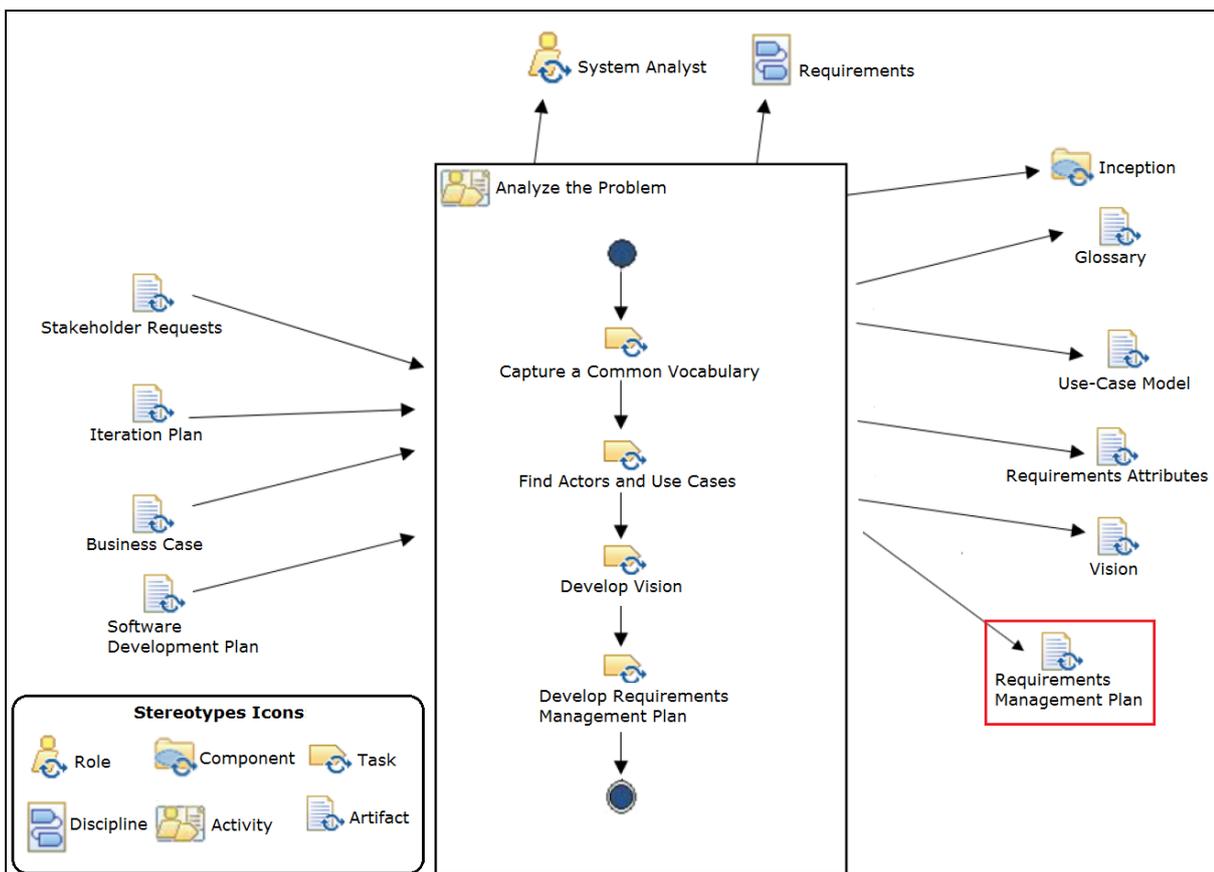


Por exemplo, para o componente *Inception* podem ser recuperadas as atividades *Analyze the Problem* (da disciplina de Requisitos), *Plan Project Configuration & Change*

Control e Create Project Configuration Management (CM) Environments, ambas da disciplina de Configuração e Gerenciamento de Mudanças. Sendo assim, o engenheiro de processos pode selecionar os componentes (atividades) mais adequados às necessidades do projeto para compor o processo de software.

Para melhor entendimento de quais elementos compõem uma atividade, a Figura 13 mostra a atividade *Analyze the Problem* e seus respectivos elementos de processo. O objetivo desta atividade é obter o acordo sobre o problema a ser resolvido, identificar as partes envolvidas, definindo os limites e identificando as restrições impostas pelo sistema (IBM RATIONAL, 2003). As demais atividades, juntamente com seus elementos são encontradas no Apêndice B.

Figura 13 – Atividade *Analyze the Problem* e elementos de processo



A Figura 13 apresenta todos os elementos de processos de uma atividade, são eles: papel, disciplina, componente, tarefas e artefatos de entrada e saída. A atividade *Analyze the Problem* da disciplina de *Requirements* do RUP, selecionada para ser instanciada no componente *Inception*, contém as tarefas *Capture a Common Vocabulary*, *Find Actors and Use Cases*, *Develop Vision* e *Develop Requirements Management Plan* realizada pelo *System*

Analyst. Estas tarefas possuem artefatos de entrada e saída e o artefato *Requirements Management Plan* (destacado na Figura 13) foi o que apresenta maior ocorrência nas associações das práticas do CMMI com os artefatos e atividades do RUP. Além do que este artefato é um dos mais relevantes desta atividade. O plano de qualidade correspondente a ele encontra-se na Tabela 26, da Apêndice C desta dissertação.

4.2.3 Elaboração da Linha de Processo de Software

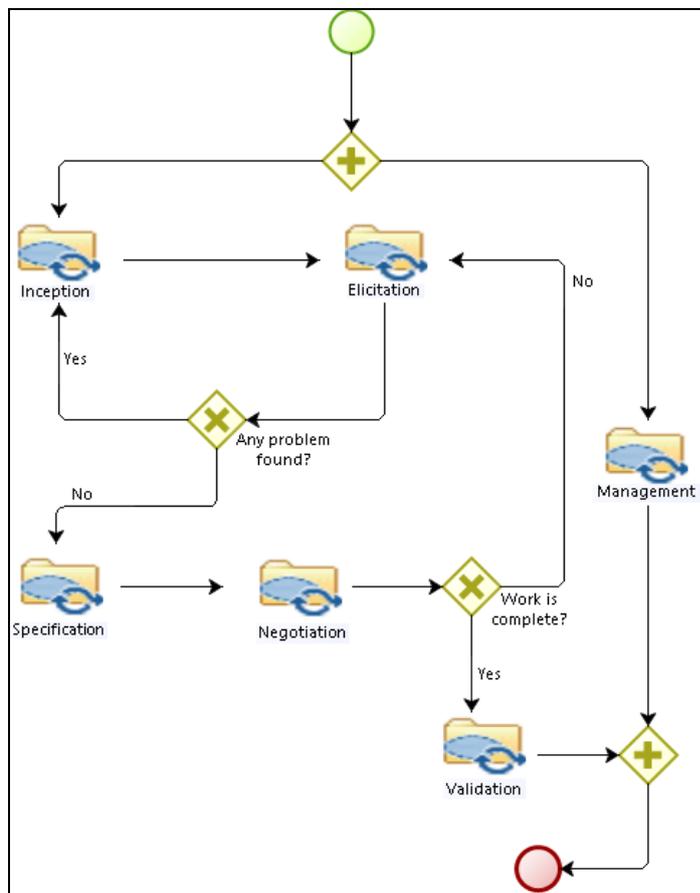
Segundo Barreto, Murta e Rocha (BARRETO; MURTA; ROCHA, 2011), uma LPrS pode ser considerada uma arquitetura de processos de software, pois possui elementos reutilizáveis, gerando processos de software distintos. Desta forma, a partir da arquitetura de processos definida, apresentada na Seção 4.2.2, foi elaborada a LPrS definida para as disciplinas de Requisitos e Configuração e Gerenciamento de Mudanças, do *framework* RUP (IBM RATIONAL, 2003).

A Figura 14 mostra que a LPrS inicia com o componente *Inception*, passa pelo componente *Elicitation*, *Specification*, *Negotiation* e *Validation*. Paralelamente a esses componentes citados está o componente *Management*. Nesse trabalho, componente refere-se a uma unidade de definição de processos e este pode ser usado em diferentes processos. Um componente reúne elementos como papéis, atividades, artefatos, disciplinas, entre outros, necessários a realização do trabalho em um processo de software (BARRETO; MURTA; ROCHA, 2011).

Para cada um destes componentes são recuperadas uma ou mais atividades que possuem caracterizações semelhantes, ou seja, que contenham contexto situacional similar. Por exemplo, para o componente *Management* tem-se as seguintes atividades: *Manage Changing Requirements* (da disciplina de Requisitos do RUP), *Monitor & Report Configuration Status*, *Manage Change Requests* e *Manage Baselines & Releases* (da disciplina de Configuração e Gerenciamento de Mudanças do RUP).

Desse modo, após as atividades serem recuperadas, priorizadas e selecionadas – utilizando os métodos matemáticos AHP, TODIM e TOPSIS – tem-se a elaboração da LPrS adaptada. Vale ressaltar que a LPrS oferece a habilidade de adaptar o comportamento do processo às mudanças das características situacionais de cada projeto e aos requisitos de adaptação selecionados. De forma flexível, a linha permite ativar ou desativar alternativas de componentes de processos dependendo do contexto do projeto. Isso permite que o recorte do processo satisfaça às mudanças nas necessidades do projeto.

Figura 14 – Linha de Processo de Software definida para a disciplina de Requisitos e Configuração e Gerenciamento de Mudanças



A LPrS criada foi utilizada para definir o processo padrão, e assim, a partir dessa linha é realizada a customização/adaptação de uma nova instância de processo para um determinado projeto. Deste modo, é possível diminuir o tempo e o esforço necessário para criar o processo de um projeto quando se aproveita um processo reutilizável, ao invés de criá-lo desde o início.

4.3 PROCESSO DE ELABORAÇÃO DOS PLANOS DE QUALIDADE

Visando resolver uma das limitações apresentadas na dissertação de Bertuol (BERTUOL, 2014), que está no fato da escolha dos elementos que compõem o plano de avaliação dos artefatos ser fortemente dependente da intervenção humana, uma das propostas deste trabalho é uma abordagem semi-automatizada desse processo. Além disso, o uso de modelos de avaliação de processos de software é bem difundido nesta área e esses modelos pressupõem que a avaliação seja executada desde o início do processo de desenvolvimento por meio da avaliação de artefatos intermediários. Como já informado anteriormente, optou-se

neste trabalho por utilizar o CMMI (SEI, 2010) como modelo de avaliação de processos, os elementos do RUP para associar com as práticas e as características de avaliação são descritas a partir da norma ISO/IEC 9126 (ISO/IEC 9126, 2003).

O processo de avaliação foi construído com base na norma ISO/IEC 14598 (ISO/IEC 14598, 1999). A escolha dessa norma deu-se tanto pela sua compatibilidade com os conceitos estabelecidos pelos modelos de qualidade quanto por descrever um processo de avaliação das características de qualidade descritas pela ISO/IEC 9126 (ISO/IEC 9126, 2003), sendo ambas incorporadas à ISO/IEC 25000 (*SQuaRE*) (ISO/IEC 25000, 2005), o que a torna um modelo consistente e amplamente referenciado na comunidade acadêmica.

A sistemática para elaboração do plano de qualidade iniciou a partir da interpretação e análise das áreas de processo Gerenciamento de Requisitos e Gerenciamento de Configuração, propostas no CMMI nível 2. As práticas descritas pelo CMMI foram relacionadas com atividades e artefatos descritos no RUP que visam satisfazer a prática.

As atividades que satisfazem determinadas práticas selecionadas serão recuperadas e priorizadas, então se forem escolhidas, irão compor o processo adaptado. Após a criação da LPrS, um plano será gerado contendo os elementos de qualidade que servirão para avaliar os artefatos das atividades do processo. No entanto, o engenheiro de processos poderá realizar alterações no mesmo.

Como primeiro passo para determinar o plano de qualidade, foi necessário povoar o repositório. Os dados foram provenientes da literatura especializada, incluindo os modelos e trabalhos específicos. Posteriormente, este repositório poderá ser expandido pela experiência de especialistas em projetos desenvolvidos anteriormente. Em seguida, foi necessário definir os elementos de qualidade – apresentados no metamodelo – composto pelos seguintes itens: *Artefato, Objetivo, Meta de Qualidade, Submeta de Qualidade, Tipo de Qualidade, Método de Avaliação, Métrica e Limite*. Para cada item existem instâncias já cadastradas e a possibilidade de manutenção de cada uma delas ou inserção de novas.

Alguns subprocessos da ISO/IEC 14598 foram ajustados de forma que pudessem refletir aspectos específicos da avaliação de artefatos de software. Todavia, a sequência da avaliação ficou semelhante: i – estabelecer requisitos de avaliação (a definição do que se quer avaliar); ii – especificar a avaliação (a seleção de metas (ou características) de qualidade relacionadas ao objeto em avaliação); iii – projetar a avaliação (produzir o plano de avaliação). A última etapa da ISO/IEC 14598 é executar a avaliação, que consiste na inspeção, medição e teste dos produtos e seus componentes de acordo com o plano de

avaliação. Esta etapa está fora do escopo deste trabalho e fica a cargo de pesquisas futuras ou de possíveis usuários da abordagem.

Outro ajuste da ISO/IEC 14598 foi incluir um passo na etapa 2 responsável pela especificação da avaliação, que é a de relacionar as metas de qualidade da ISO/IEC 9126 com os respectivos métodos de avaliação, as métricas, os valores limites e as práticas do CMMI para cada artefato correspondente. Vale lembrar que, a definição de valores limites para as métricas não é obrigatório, além disso, os limites definidos nos cadastros das métricas são usados como referência aqui, na especificação da avaliação, mas o avaliador pode alterar esses valores para cada especificação de avaliação.

A etapa 3 compreende na documentação dos procedimentos definidos e que serão usados pelo avaliador para definir a qualidade dos artefatos selecionados, ou seja, nesta fase é produzido o plano de qualidade.

Ao finalizar a etapa 3, entende-se que o armazenamento de dados referentes aos planos de qualidade definidos durante o processo de avaliação para um projeto específico possa servir como referência para guiar a avaliação em projetos futuros, visto que os artefatos gerados durante um processo de software são semelhantes se mantidas as mesmas abordagens de desenvolvimento e os mesmos modelos de processo de software.

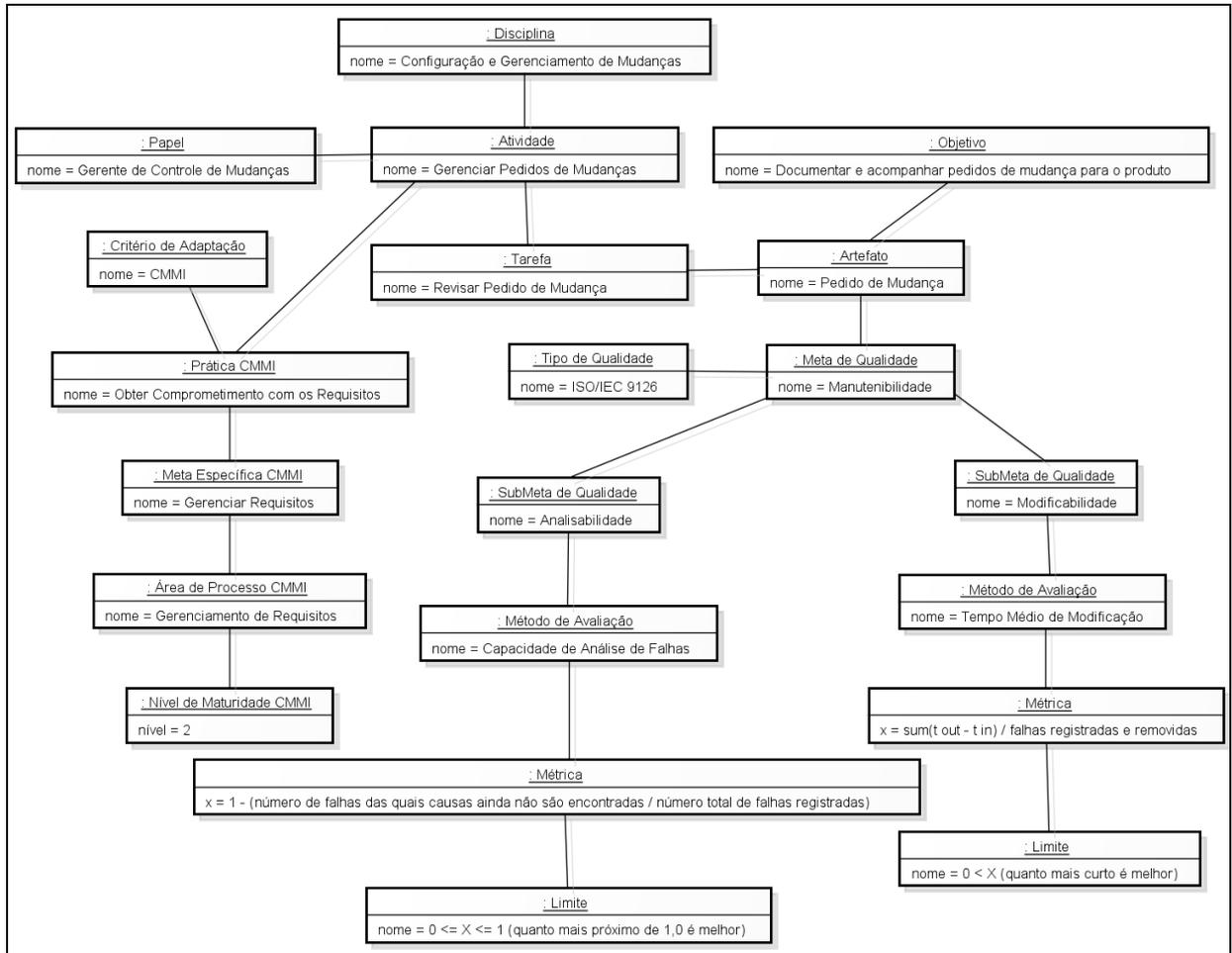
Para exemplificar a estruturação do plano de qualidade e o funcionamento da abordagem, vamos considerar que o usuário tenha selecionado a prática do CMMI “*Obter Comprometimento com os Requisitos*”, esta prática pertence a Meta Específica “*Gerenciar Requisitos*”, que faz parte da Área de Processo “*Gerenciamento de Requisitos*” descrita no nível 2 de maturidade do CMMI. Esta prática possui como Produto de Trabalho Típico “*Acordos documentados sobre os requisitos e suas mudanças*”.

Ao selecionar estes requisitos de adaptação, a ferramenta irá realizar a busca entre as atividades do repositório que satisfazem essa prática. Por exemplo as atividades “*Gerenciar Pedidos de Mudanças*” e “*Gerenciar Mudanças de Requisitos*” das disciplinas do RUP de *Configuração e Gerenciamento de Mudanças e Requisitos*, respectivamente, serão recuperadas e priorizadas. Após a priorização das atividades e se esta atividade for selecionada para compor a LPrS adaptada, o objetivo então será realizar a avaliação da qualidade dos artefatos das atividades do processo.

Cada atividade possui um ou mais artefatos de entrada e saída, e este conjunto de artefatos serão vinculados a um plano de qualidade que descreve como avaliá-los, considerando, sobretudo, suas características, propósitos, interessados, métodos de avaliação e métricas; isso tudo com vistas à melhoria da qualidade do produto final.

A Figura 15 mostra a instanciação do metamodelo para a avaliação do artefato “*Pedido de Mudança*” da atividade “*Gerenciar Pedidos de Mudanças*” do RUP.

Figura 15 – Instância do QAPro-M



O propósito deste artefato é documentar e acompanhar pedidos de mudanças para o produto. A meta de qualidade principal é a *manutenibilidade*, baseada na qualidade interna proposta pela ISO/IEC 9126. A partir dela são exploradas as submetas *analisabilidade* e *modificabilidade*. Essas metas e submetas podem ser avaliadas por métodos específicos e quantificadas por métricas relacionadas.

A Tabela 3 mostra o plano de qualidade para este mesmo artefato. Os planos elaborados foram organizados e divididos em três partes. Primeiramente são mostrados os dados do CMMI, com a área de processo, o nível de maturidade, a meta específica e a prática específica. A segunda parte está relacionada aos dados do RUP, no qual possui disciplina, atividade, tarefa e papel que estão vinculados. Por fim, encontram-se os elementos de qualidade responsáveis pela avaliação do artefato em questão.

Tabela 3 – Plano de Qualidade: Artefato Pedido de Mudança

PLANO DE QUALIDADE		
Dados do CMMI		
Área de Processo:	Gerenciamento de Requisitos	Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos	
Prática Específica:	SP 1.2 Obter Comprometimento com os Requisitos SP 1.3 Gerenciar Mudanças nos Requisitos SP 1.5 Identificar Inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos	
Dados do RUP		
Disciplina:	Configuração e Gerenciamento de Mudanças	
Atividade:	Gerenciar Pedidos de Mudanças	
Tarefa:	Revisar Pedido de Mudança	
Papel:	Gerente de Controle de Mudanças	
Avaliação do Artefato		
Artefato:	Pedido de Mudança	
Objetivo:	Documentar e acompanhar pedidos de mudança para o produto.	
Meta de Qualidade:	Manutenibilidade	Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Analisabilidade	
Método de Avaliação:	Capacidade de análise de falhas	
Métrica:	$X = 1 - (\text{número de falhas das quais causas ainda não são encontradas} / \text{número total de falhas registradas})$	
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)	
SubMeta de Qualidade:	Modificabilidade	
Método de Avaliação:	Tempo médio de modificação	
Métrica:	$X = \text{sum}(t_{\text{out}} - t_{\text{in}}) / \text{falhas registradas e removidas}$	
Limite:	$0 < X$ (quanto mais curto é melhor)	

Enquanto que a Tabela 4 utiliza os mesmos dados do RUP e da avaliação do artefato em questão, porém, o que diferencia são as práticas do CMMI que estão relacionadas à área de processo Gerenciamento de Configuração. Para exemplificar, a prática SP 2.2 denominada “*Controlar Itens de Configuração*” da meta SG 2 “*Acompanhar e Controlar Mudanças*” possui a subprática “*Registrar as mudanças nos itens de configuração e os motivos das mudanças, conforme apropriado*”, ou seja, esta subprática está relacionada ao artefato “*Pedido de Mudança*”. Portanto, este artefato está relacionado tanto para a área de processo do CMMI Gerenciamento de Requisitos quanto para Gerenciamento de Configuração.

Tabela 4 – Dados do CMMI para as Práticas da Área de Processo Gerenciamento de Configuração

Dados do CMMI		
Área de Processo:	Gerenciamento de Configuração	Nível de Maturidade: 2
Meta Específica:	SG 1. Estabelecer Baselines SG 2. Acompanhar e Controlar Mudanças	
Prática Específica:	SP 1.2 Estabelecer um Sistema de Gestão de Configuração SP 2.1 Acompanhar Solicitações de Mudança SP 2.2 Controlar Itens de Configuração	

A ideia principal dos planos de qualidade é organizar os dados, estruturados pelo metamodelo, de forma que os avaliadores tenham uma referência sólida, clara e

compreensível no momento de avaliar os artefatos produzidos durante o projeto com base nas metas de qualidade que julgam serem as mais importantes para o produto final.

Os demais planos de qualidade encontram-se no Apêndice C, no entanto, vale ressaltar que os planos de qualidade elaborados são sugestões deste trabalho com base na literatura.

Resumidamente, para cada um dos componentes da LPrS podem ser selecionadas diferentes atividades que contenham contexto situacional similar e que contemplem as práticas propostas pelo CMMI que se deseja implantar no processo adaptado. Estas atividades possuem artefatos e estes são associados a planos de qualidade. Deve-se salientar que o plano de qualidade gerado servirá como referência para os engenheiros de processo, no entanto, os mesmos poderão alterá-los caso seja necessário.

4.3.1 Definição de Métodos de Avaliação e Métricas

Para avaliar a qualidade de software é necessário que haja uma maneira de medi-la, no entanto, as características de qualidade proposta pela ISO/IEC 9126 – descrita na Seção 2.1 – não deixam claro as métricas necessárias para essa medição. Portanto, neste trabalho foi preciso estabelecer estas métricas e seus respectivos métodos de avaliação, a fim de correlacionar estes às características de qualidade.

Vale ressaltar que para encontrar as métricas que correspondem a determinada prática, foi analisado o CMMI, a fim de procurar possíveis características de qualidade para esta associação. No Apêndice D é apresentado um conjunto de métricas que foram retiradas da literatura e da ISO/IEC 9126, algumas delas foram adaptadas conforme fosse necessário.

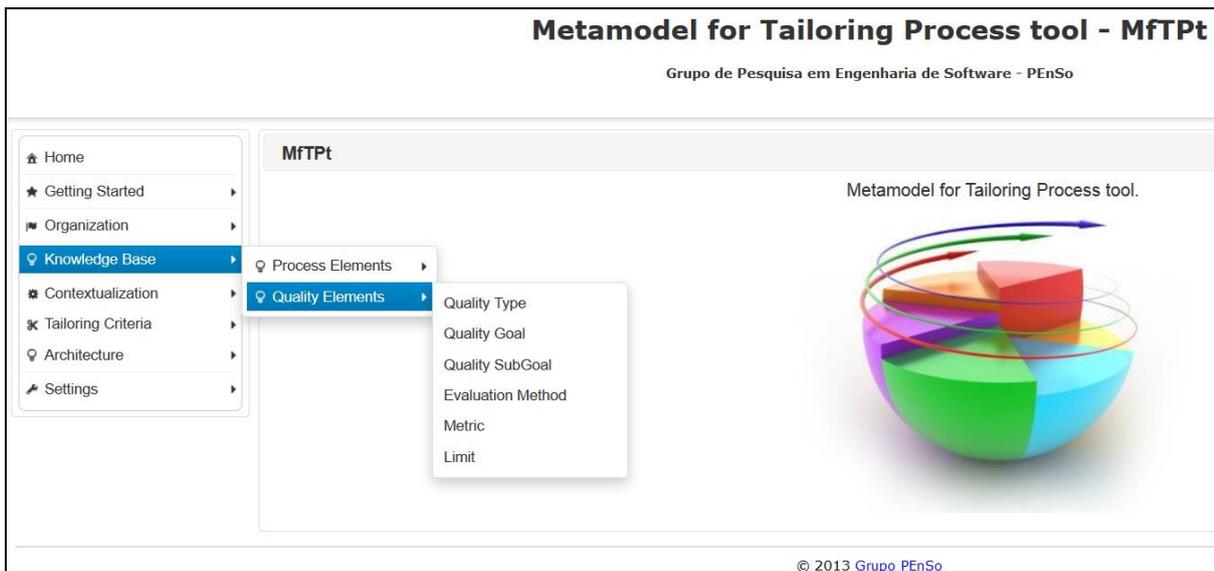
4.4 FERRAMENTA DE APOIO *QAPRO SYSTEM*

Nesta seção alguns formulários da ferramenta, denominada *Quality Assessment of Artifacts in Process (QAPro System)*, são mostrados a fim de ilustrar a implementação do *framework* de avaliação de artefatos de software. A *QAPro System* foi desenvolvida a partir de uma extensão da ferramenta MfTPt (LORENZ, 2014). Essa foi desenvolvida na linguagem Java para especificação das regras de negócio, *Java Persistence API (JPA)* e *Hibernate* para persistência e acesso a dados; *Java Server Faces (JSF)*, HTML, CSS3 e *JQuery* para compor a camada de apresentação e o banco de dados utilizado foi o *MySQL*.

Na Figura 16 é ilustrada a tela inicial da ferramenta, tendo à esquerda o menu de opções para a navegação das principais funcionalidades do sistema. Na opção de base de

conhecimento (*Knowledge Base*) possui os elementos de qualidade (*Quality Elements*) cadastrados. Este é o lugar onde os usuários terão acesso à inserção e manutenção dos dados, composto pelos seguintes itens: *Quality Type*, *Quality Goal*, *Quality SubGoal*, *Evaluation Method*, *Metric* e *Limit*. Já os elementos de processo (*Process Elements*) contém os itens: *Action*, *Activity*, *Artifact*, *Discipline*, *Life Cycle*, *Phase*, *Role*, *Source* e *Task*.

Figura 16 – Tela Principal do Sistema com Elementos de Qualidade



Além da categoria *Knowledge Base*, o menu principal é subdividido da seguinte forma: *Getting Started*, *Organization*, *Contextualization*, *Tailoring Criteria*, *Architecture* e *Settings*. Detalhes de cada categoria encontram-se na dissertação de Lorenz (LORENZ, 2014).

Nas Figuras 17 e 18 são apresentadas, respectivamente, a tela de cadastro e listagem para o elemento *Quality Goal*, na qual são cadastradas e listadas propriedades como nome, descrição e tipo de qualidade.

Figura 17 – Exemplo de Cadastro do Elemento *Quality Goal*

Create a quality goal

Name

Description

Quality type

Back Save Reset Delete

Figura 18 – Exemplo de Listagem do Elemento *Quality Goal*

Quality Goals			
 << < 1 2 3 > >> 10 			
Id	Name	Description	Quality type
1	Functionality	The capability of the software to provide functions which meet the stated and implied needs of users under specified conditions of usage (what the software does to meet needs).	External Quality - ISO/IEC 9126
2	Reliability	The capability of the software product to maintain its level of performance under stated conditions for a stated period of time.	External Quality - ISO/IEC 9126
3	Usability	The capability of the software product to be understood, learned, used and provide visual appeal, under specified conditions of usage (the effort needed for use) .	External Quality - ISO/IEC 9126
4	Efficiency	The capability of the software product to provide desired performance, relative to the amount of resources used, under stated conditions.	External Quality - ISO/IEC 9126
5	Maintainability	The capability of the software product to be modified which may include corrections, improvements or adaptations of the software to changes in the environment and in the requirements and functional specifications (the effort needed for modification).	Internal Quality - ISO/IEC 9126
6	Portability	The capability of the software product to be transferred from one environment to another. The environment may include organizational, hardware or software.	Internal Quality - ISO/IEC 9126
7	Effectiveness		Quality in Use - ISO/IEC 9126
8	Productivity		Quality in Use - ISO/IEC 9126

Sendo assim, a ferramenta desenvolvida estabelece uma série de formulários nos quais o responsável pela avaliação da qualidade pode inserir e alterar os dados referentes aos elementos que compreendem o processo de avaliação.

Para o cadastramento das atividades, além dos atributos básicos como nome, descrição, propósito, disciplina e papel, é necessário selecionar as tarefas que pertencem a respectiva atividade e em seguida associá-la aos seus respectivos artefatos. Cada atividade tem uma ou mais tarefas associadas e cada tarefa possui um ou mais artefatos. Por exemplo, para a atividade “*Manage Change Requests*” tem-se a tarefa “*Review Change Requests*”, cujo artefato associado é o “*Change Request*”.

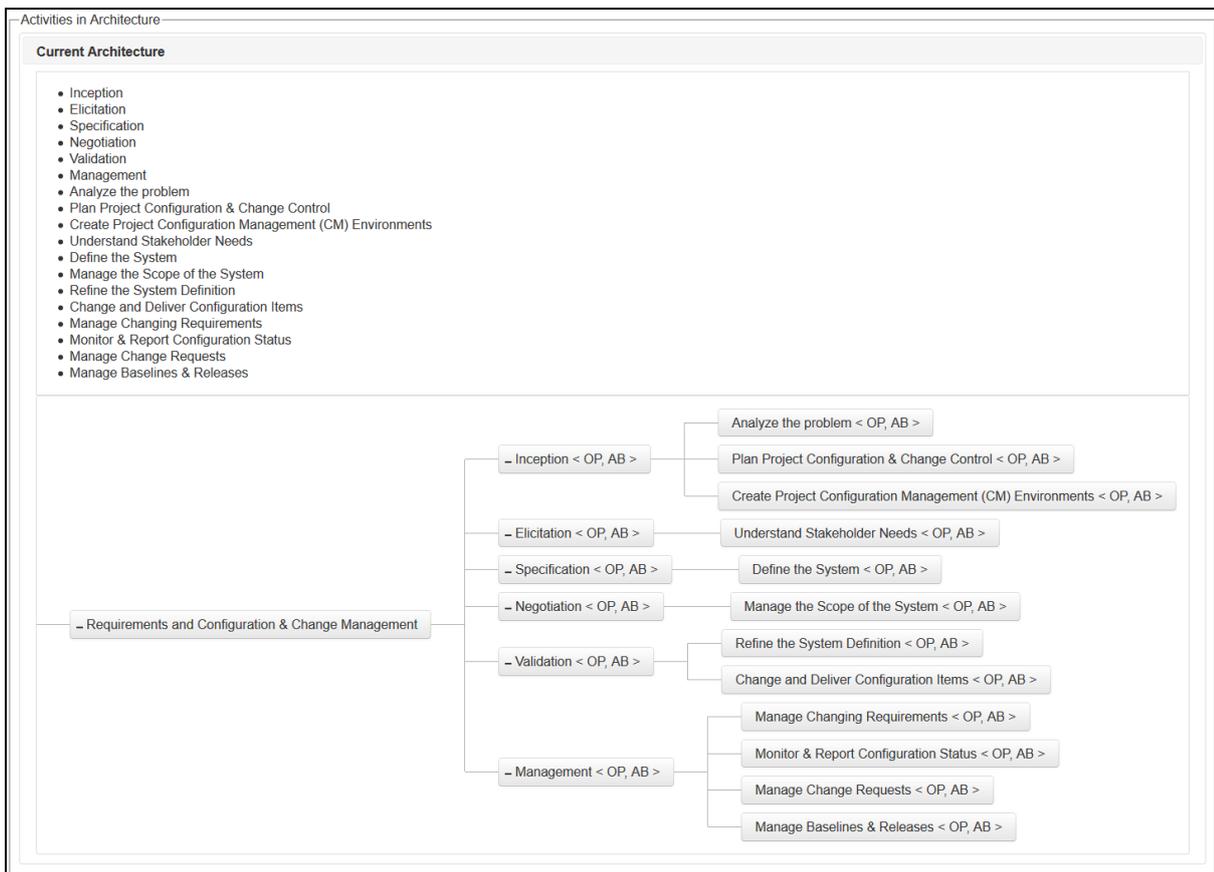
Além disso, para cada atividade deve-se definir contextos e requisitos de adaptação. A contextualização utilizada foi o *Octopus Model*, definindo valores para cada um dos seus oito fatores para a atividade em questão. Ainda é necessário definir qual arquitetura a atividade pertence, sendo que ela pode pertencer a mais de uma arquitetura.

Na etapa 2 da adaptação de processos de software baseado em LPrS, como já mencionado na Seção 3.2, é a etapa da seleção dos requisitos de adaptação e da arquitetura de processos que devem ser considerados no processo adaptado. Para isso foi necessário realizar o cadastro da arquitetura e dos requisitos de adaptação que foram usados nesse trabalho.

Na opção *Architecture* do menu principal, o usuário tem a listagem das diferentes arquiteturas cadastradas e a opção de cadastramento de uma nova arquitetura. Deste modo,

para esta pesquisa, a arquitetura definida e cadastrada na ferramenta foi a descrita na Seção 4.2.2, nomeada de *Requirements and Configuration & Change Management*. Na Figura 19 está apresentada a arquitetura definida com suas respectivas atividades.

Figura 19 – Arquitetura *Requirements and Configuration & Change Management*



Para a definição da arquitetura pode-se escolher a variabilidade da estrutura por meio do uso de componentes. Os componentes obrigatórios (*mandatory-M*) são usados para definir os elementos que devem estar presentes em todos os processos organizacionais, isto é, o conjunto mínimo de elementos a serem executados em um processo. Componentes opcionais (*optional-OP*) definem elementos que podem ou não compor o processo organizacional.

Além disso, os componentes podem ser concretos (*concrete-C*) ou abstratos (*abstract-AB*). Um componente concreto não permite qualquer tipo de variabilidade. Um componente abstrato é uma maneira de representar as variabilidades, sendo considerado um ponto de variação. Ele pode ser diretamente substituído por componentes concretos ou componentes abstratos, formando hierarquias de componentes abstratos, que em algum momento precisam ser substituídos por componentes concretos. Para esta arquitetura foram escolhidos todos os componentes opcionais, para que seja possível selecionar apenas aqueles que interessam para

o projeto em questão, ou seja, pode-se ou não utilizá-lo no processo e abstratos, no qual admite variabilidades, não estando vinculado a uma única forma de realização.

Posteriormente, como já comentado anteriormente, cada atividade pode-se definir um ou mais requisitos de adaptação, os tipos de requisitos já cadastrados na ferramenta são riscos e segurança.

Deste modo, a categoria *Tailoring Criteria* possui a opção *Criteria Type*, no qual foi cadastrado o novo tipo de requisito de adaptação, relacionado a área de qualidade. Os requisitos adicionados foram as práticas específicas (*Specific Practices - SP*) da área de processo *Requirements Management* e *Configuration Management*, ambos do nível 2 do CMMI, como mostra a Figura 20.

Figura 20 – Requisitos de Adaptação: CMMI

CMMI - Requirements Management: Level 2	
<input type="checkbox"/> SP 1.1 Understand Requirements	<input type="checkbox"/> SP 1.2 Obtain Commitment to Requirements
<input type="checkbox"/> SP 1.3 Manage Requirements Changes	<input type="checkbox"/> SP 1.4 Maintain Bidirectional Traceability of Requirements
<input type="checkbox"/> SP 1.5 Ensure Alignment Between Project Work and Requirements	
CMMI - Configuration Management: Level 2	
<input type="checkbox"/> SP 1.1 Identify Configuration Items	<input type="checkbox"/> SP 1.2 Establish a Configuration Management System
<input type="checkbox"/> SP 1.3 Create or Release Baselines	<input type="checkbox"/> SP 2.1 Track Change Requests
<input type="checkbox"/> SP 2.2 Control Configuration Items	<input type="checkbox"/> SP 3.1 Establish Configuration Management Records
<input type="checkbox"/> SP 3.2 Perform Configuration Audits	

Por fim, como mostra o processo da Figura 10, a última tarefa corresponde a gerar plano para avaliar a qualidade dos artefatos do processo criado. Para que os planos de qualidade sejam gerados de acordo com a abordagem proposta é necessário que os elementos de qualidade tenham sido cadastrados/definidos e armazenados no repositório. Os elementos de qualidade são instâncias das classes descritas no metamodelo QAPro-M, apresentado na Seção 4.1.

Esta etapa é realizada da seguinte forma: para cada atividade do processo criado, é selecionada a tarefa dessa respectiva atividade e, em seguida, seleciona o artefato que deseja obter o plano de avaliação da qualidade. A Figura 21 mostra que o artefato *Software Requirements Specification* da tarefa *Capture a Common Vocabulary*, pertencente a atividade *Define the System* foi selecionado para a geração de seu plano de avaliação de qualidade.

Figura 21 – Detalhes Iniciais do Plano de Avaliação de Qualidade de Artefato

Initial details of your Quality Plan

Quality Plan: RUP

Activity:

Task:

Artifact:

Para cada artefato selecionado é apresentada a especificação da avaliação, com o seu objetivo, suas metas e submetas de qualidade, tipo de qualidade, método de avaliação, assim como a métrica e o possível valor limite utilizados para a avaliação deste artefato. Na Figura 22 é ilustrado um exemplo de um plano de avaliação da qualidade gerado para determinado artefato que pertence a uma atividade que compõe a LPrS adaptada. Estes planos de qualidade podem ser exportados nos seguintes formatos: excel, pdf, csv ou xml.

Esta documentação é a que será usada pelo analista de qualidade para definir a qualidade dos artefatos selecionados. A ideia do plano é que todos os elementos de qualidade estejam estruturados de forma clara e compreensível para que sirva como um guia aos avaliadores na avaliação da qualidade dos artefatos do processo.

Figura 22 – Exemplo de um Plano de Avaliação de Qualidade de Artefato

Quality Plans								
Id	Artifact	Description	Quality Type	Quality Goal	Quality SubGoal	Evaluation Method	Metric	Limit
19	Software Requirements Specification	This artifact captures the software requirements for the complete system, or a portion of that system.	External Quality - ISO/IEC 9126	Functionality	Functionality compliance	Functional compliance	X=1-(number of functionality compliance items specified that have not been implemented during testing / total number of functionality compliance items specified)	0<=X<=1 The closer to 1.0 is the better.
20	Software Requirements Specification	This artifact captures the software requirements for the complete system, or a portion of that system.	External Quality - ISO/IEC 9126	Usability	Understandability	Completeness of description	X=number of requirements described / total number of system requirements	0<=X<=1 The closer to 1.0 is the better.

No entanto, os avaliadores têm a liberdade para alterar qualquer elemento de qualidade, conforme seja mais apropriado para avaliar a qualidade dos artefatos. Dessa maneira, os requisitos de flexibilidade e transparência são mantidos durante o processo

avaliativo. Na Figura 23 mostra a tela de edição do plano de avaliação de qualidade do artefato *Software Requirements Specification*.

Figura 23 – Edição de um Plano de Avaliação de Qualidade de Artefato

The screenshot displays a web-based form titled "Edit a quality plan". The form contains the following fields and controls:

- Artifact:** A dropdown menu with "Software Requirements Specification" selected.
- Description:** A text area containing the text: "This artifact captures the software requirements for the complete system, or a portion of that system."
- Quality Type:** A dropdown menu with "External Quality - ISO/IEC 9126" selected.
- Quality Goal:** A dropdown menu with "Usability" selected.
- Quality SubGoal:** A dropdown menu with "Understandability" selected.
- Evaluation Method:** A dropdown menu with "Completeness of description" selected.
- Metric:** A dropdown menu with "X=number of requirements described / total number of system requirements" selected.
- Limit:** A dropdown menu with "0<=X<=1 The closer to 1.0 is the better." selected.

At the bottom of the form, there are four buttons: "Back", "Save", "Reset", and "Delete".

Resumidamente, o desenvolvimento da *QAPro System* a partir da extensão da ferramenta MfTPt permite que seja definido um plano de qualidade para cada artefato com base nas atividades do processo gerado, considerando uma série de aspectos, como: o objetivo desse artefato, as metas de qualidade, o método de avaliação a ser utilizado, a métrica, entre outros.

5 TRABALHOS RELACIONADOS

Este capítulo destaca alguns trabalhos que compartilham das definições apresentadas nesta dissertação. Nesse contexto, destacam-se os trabalhos de (RODRÍGUEZ et al., 2010), (VANDERROSE; HABRA, 2011), (LEE; LEE, 2005), (BERNABÉ et al., 2011) e (STORCH; LAUE; GRUHN, 2013).

Rodríguez *et al.* (RODRÍGUEZ et al., 2010) defendem que não existe uma metodologia genérica e flexível que permita avaliar a qualidade para todos os tipos de artefatos de software, muito menos uma ferramenta que suporte isso. Como solução, os autores propõem um ambiente denominado *Continuous Quality Assessment (CQA)*, que é composto por uma metodologia (*CQA-Meth*) e uma ferramenta (*CQA-Tool*). O objetivo principal do *CQA-Meth* consiste em definir um *framework* que permita estabelecer os processos necessários para a realização da avaliação de qualquer tipo de artefatos de software, bem como para tornar possível a comunicação entre o cliente e a equipe de avaliação. O *CQA-Tool* oferece a facilidade para a construção de um catálogo de técnicas de avaliação que integra as técnicas disponíveis para cada artefato de software (por exemplo métricas, *checklists*, convenções de modelagem, guias, etc.).

O processo de avaliação é realizado em 4 fases: planejamento (produzir um contrato e obter um plano para avaliação de modelos UML), especificação (definir o escopo da avaliação, definindo o conjunto de características que vão ser avaliados, as técnicas e ferramentas que vão ser usadas), execução (aplicar as técnicas de avaliação) e por fim, a conclusão (relatar a avaliação e apresentar os resultados).

Vanderose e Habra (VANDERROSE; HABRA, 2011) propõem um *framework* com o objetivo de avaliar artefatos de software. A abordagem é um modelo integrado de avaliação capaz de representar os vários fatores de qualidade para cada projeto, além dos relacionamentos entre esses fatores e os diversos métodos de avaliação usados na medição das entidades. Porém, em vez de determinar modelos de qualidade específicos para vários produtos de software, o objetivo é que o *framework* proposto ajude na construção de um modelo que concentre os esforços no planejamento necessário à avaliação dos artefatos do projeto.

A semelhança em relação este trabalho está no fato de que para estruturar a proposta, os autores especificam um metamodelo que captura os conceitos dos modelos tradicionais como a ISO/IEC 9126 e alinham métodos de avaliação aos interesses dos vários *stakeholders*

envolvidos. A metodologia de avaliação, por sua vez, é baseada nos princípios da abordagem *Goal Question Metric* (GQM) (BASILI; CALDEIRA; ROMBACH, 1994) e inclui cinco passos: i – capturar as informações contextuais (*stakeholders*, motivações, tempo, orçamento); ii – inserir as informações coletadas na estrutura do metamodelo; iii – definir o plano de avaliação genérico; iv – submeter os artefatos selecionados ao plano de avaliação; e, v – interpretar os resultados.

A abordagem de Lee e Lee (LEE; LEE, 2005) afirmam que as normas ISO/IEC 9126 e ISO/IEC 14598 não fornecem orientações práticas para aplicar o modelo de qualidade e o processo de avaliação em projetos reais. Deste modo, este trabalho apresenta um modelo de avaliação de qualidade de software quantitativo para os artefatos da metodologia de Desenvolvimento Baseado em Componentes. Particularmente, o modelo proposto adota a ISO/IEC 9126 com a aplicação de pesos nas características de qualidade, estes são obtidos por meio de questionários selecionados pelos *stakeholders* e pelo método matemático AHP.

Foram aplicados 50 questionários, dentre os entrevistados estão: 17 gerentes, 12 desenvolvedores, 11 usuários e 10 engenheiros de manutenção. Eles trabalham em organizações governamentais, institutos de pesquisa e empresas. Foi investigado também o grau de reconhecimento dos entrevistados sobre a importância da avaliação da qualidade e da adequação da avaliação da qualidade utilizando características de qualidade. Como resultado, 44 entrevistados (88%) responderam que a avaliação de qualidade de software é importante.

Bernabé *et al.* (BERNABÉ *et al.*, 2011) propõem um modelo de qualidade de dados que pode ser usado como referência por gerentes de projeto para avaliar e, se necessário, melhorar a qualidade dos valores de dados correspondente aos metadados, descrevendo os artefatos utilizados no processo de planejamento de um projeto de desenvolvimento de software. Foi utilizado os artefatos descritos na norma ISO/IEC 12207 (ISO/IEC 12207, 2008), na parte de planejamento de processo. Então, compararam esses artefatos encontrados com os propostos pelo *Project Management Body of Knowledge* (PMBOK) (PROJECT MANAGEMENT INSTITUTE - PMI, 2013), a fim para descrever melhor a sua estrutura e construir o modelo de qualidade de dados.

Storch *et al.* (STORCH; LAUE; GRUHN, 2013) motivados por um modelo de qualidade de software padrão, os autores discutiram as características e subcaracterísticas de modelos de qualidade e sugeriram medidas para as mesmas. Além disso, ampliaram a ferramenta *SonarQube*, esta possui diferentes medidas para a qualidade do software de tal forma que agora pode ser usado com repositórios de modelos de processos de negócio também. Isto permite avaliar a qualidade de uma coleção de modelos, da mesma forma que já

está bem estabelecido para avaliar a qualidade do código de software. Os autores acreditam que se o controle de qualidade for realizado com antecedência, isto pode levar à detecção de possíveis problemas nas fases iniciais do processo de desenvolvimento de software.

Este trabalho mostra como estabelecer um *framework* para agregar e visualizar relatórios de qualidade do software. Desta forma, é possível monitorar o estado e o progresso da qualidade em um projeto de software – a começar nas primeiras fases antes que uma única linha de código seja escrita. O foco deste artigo não é para desenvolver novos métodos para medir o modelo de qualidade, mas para tornar os resultados das ferramentas disponíveis em uma visão agregada. Então fazem uso de controles de qualidade existentes para os diferentes aspectos do modelo de qualidade. Esses controles de qualidade já foram implementadas em várias ferramentas de modelagem. Com isso, o objetivo do trabalho foi resumir e classificar os resultados dessas ferramentas existentes em vários pontos de vista. E assim, ajudam um revisor para obter uma primeira impressão sobre a qualidade de uma coleção de modelos.

5.1 ANÁLISE DOS TRABALHOS RELACIONADOS

Com o objetivo de enfatizar as semelhanças e diferenças, a Tabela 5 mostra o comparativo entre a abordagem proposta neste trabalho e os trabalhos relacionados. Deste modo foram avaliados sete critérios de comparação:

- Tipos de Artefatos – se abrange todos os tipos de artefatos gerados em um processo.
- *Framework* – se o trabalho relaciona a abordagem como um *framework*.
- Metamodelo – se foi elaborado um metamodelo para estruturar a proposta, com os elementos para o processo de avaliação.
- Modelo de Avaliação – se a base de conhecimento é construída a partir de algum modelo de avaliação.
- Processo de Avaliação – se é detalhado um processo de avaliação da qualidade de artefatos, com a elaboração de planos de qualidade.
- Ferramenta de Apoio – se apresenta uma ferramenta de apoio à abordagem que propõe.
- Adaptação de Processos – se a proposta está inserida em uma abordagem de adaptação de processos de software.

A abordagem proposta nesta dissertação, satisfaz todos os critérios citados acima.

Tabela 5 – Comparação dos Trabalhos Relacionados

Trabalhos Critérios	Rodríguez et al. (2010)	Vanderose e Habra (2011)	Lee e Lee (2005)	Bernabé et al. (2011)	Storch et al. (2013)	Abordagem proposta
Tipos de Artefatos	Modelos UML	Preferência modelos	Artefatos da metodologia de Desenvolvimento Baseado em Componentes	Artefatos gerados no planejamento do processo da ISO/IEC 12207	Coleção de modelos de processo de negócio	Genérico a vários tipos de artefatos
Framework	Sim	Sim, <i>Model-Centric Quality Assessment (MoCQA) framework</i>	Sim	Não	Sim	Sim
Metamodelo	Não	Sim	Não	Não	Não	Sim, <i>QAPro-M</i>
Modelo de Avaliação	Não	Não	Não	Sim, o PMBOK	Não	Sim, o modelo de qualidade CMMI
Processo de Avaliação	Sim, definido pelos autores	GQM	Baseado na ISO/IEC 14598	Não	Não	Baseado na ISO/IEC 14598
Ferramenta de Apoio	Sim, <i>CQA Tool</i>	Sim, <i>QuaTALOG</i>	Não	Não	Sim, <i>SonarQube</i>	Sim, <i>QAPro System</i>
Adaptação de Processos	Não	Não	Não	Não	Não	Sim, processos planejados e ágeis

6 ESTUDOS DE CASO

Para validar a proposta deste trabalho, foram realizados cinco estudos de caso envolvendo diferentes projetos reais de software de empresas distintas. Com o propósito de obter um ponto de vista de profissionais, os estudos de caso foram realizados por três gerentes de projetos (um com doutorado, um com mestrado e um com especialização), um analista de conformidade (com mestrado) e um analista de sistemas (com graduação).

O estudo de caso, no entanto, dividiu-se em 3 etapas:

- Etapa 1 – Coleta de dados do participante e do projeto: foi entregue ao participante um formulário de caracterização (encontra-se no Apêndice E) com o objetivo de caracterizar o perfil de cada participante, assim como o contexto do projeto em questão. Além disso, para agilizar o processo da etapa 2, o participante preencheu os requisitos de adaptação que pretendia satisfazer no processo. Portanto, o resultado desta etapa foi o formulário com os dados do participante, do contexto do projeto e dos requisitos de adaptação preenchidos.
- Etapa 2 – Criação do processo adaptado e planos de qualidade gerados: com os dados solicitados na etapa 1 e com apoio da ferramenta *QAPro System*, o processo foi adaptado para atender os requisitos de determinado projeto e também os planos de qualidade foram gerados para cada artefato selecionado para o processo. O resultado desta etapa foi enviado para o participante, a fim do mesmo avaliar e poder responder as questões da etapa 3.
- Etapa 3 – Entrevista com especialistas: esta etapa teve como objetivo entrevistar os especialistas a fim de validar o processo e os planos gerados. A entrevista aplicada (encontra-se no Apêndice E) foi constituída de 15 perguntas divididas em três tópicos: processos de software, qualidade de software e planos de qualidade de artefatos. A duração da entrevista foi de aproximadamente 40 minutos, variando conforme o especialista.

Inicialmente, os participantes foram contatados com antecedência por e-mail a fim de convidá-los para o estudo, analisar a disponibilidade dos mesmos, explicar as etapas da validação, além de agendar um horário para a entrevista.

Nas próximas seções são descritos, resumidamente, cada um dos cinco projetos usados na validação, assim como os resultados obtidos nos estudos de caso.

6.1 ESTUDO DE CASO 1 – PROJETO SIS-ASTROS

O projeto de Sistema Integrado de Simulação ASTROS (SIS-ASTROS) tem por objetivo a pesquisa e o desenvolvimento, pela UFSM em cooperação com o Exército Brasileiro, de um sistema de simulação virtual tático. O foco do desenvolvimento deste protótipo de simulador será à simulação virtual de treinamento da parte tática do Reconhecimento, Escolha e Ocupação de Posição (REOP). REOP é o conjunto de operações executadas com a finalidade de deslocar uma bateria ASTROS de uma Posição de Tiro, Posição de Espera, Zona de Reunião, Coluna de Marcha ou Zona de Embarque para uma outra posição a fim de iniciar ou manter o Apoio de Fogo adequado. Sua execução judiciosa permitirá obter fogos rápidos e precisos, de modo a permitir um apoio de fogo ininterrupto.

Por sua vez, este estudo de caso envolveu a professora doutora da UFSM Lisandra Fontoura, que é coordenadora do projeto SIS-ASTROS e orientadora deste trabalho. Esse primeiro estudo de caso serviu como um piloto para ajustar o método de validação do trabalho.

As informações obtidas na etapa 1 do estudo de caso foram empregadas na construção da etapa 2. Então, primeiramente, os dados repassados no formulário foram adicionados na ferramenta *QAPro System*, portanto o projeto SIS-ASTROS, assim como o seu contexto foram cadastrados. A Figura 24 ilustra o projeto e o contexto do mesmo já cadastrados.

Figura 24 – Contexto do Projeto SIS-ASTROS

The screenshot displays the 'Project Context' step of the QAPro System. At the top, a progress bar indicates five steps: 1. Project Context (Fill Project Values), 2. Tailoring Requirements (Select the requirements), 3. Prioritization of Activities (Select activities for your process), 4. Software Process Line (Fill all details), and 5. Quality Evaluation of the Artifacts (Create a Quality Plan). Step 1 is highlighted in orange.

Under 'Initial details of your process', the 'Select a project' section lists several projects with radio buttons. 'SIS-ASTROS' is selected. Other projects include Ospa Project, Google Project, Facebook Project, LaCA Project, Movie rentals, Project IFF, Project SisEvent, Automação Agrícola, Empresa Dataprev, Sistema Irriga, and Ponto Eletrônico UFSM.

The 'Contextualize your project' section shows eight Octopus Model cards, each with a title and radio button options:

- Size:** small, medium, large
- Stable Architecture:** Stable, Changed, New
- Business Model:** In house, Commercial, Large System Component
- Team Distribution:** Collocated, Different teams, Geographic
- Rate of Change:** More than 30, From 10 to 30, Less than 10
- Age of System:** New development, Maintenance, Legacy evolution
- Criticality:** Comfort loss, Essential money loss, Deaths
- Governance:** Dynamic/flexible, Simple rules, Mechanic/formal

No passo 2, denominado *Tailoring Requirements* é selecionada a arquitetura e os requisitos de adaptação para a geração do processo adaptado deste projeto em questão. A arquitetura selecionada foi *Requirements and Configuration & Change Management*, elaborada neste trabalho. Os requisitos de adaptação escolhidos pela participante para serem considerados durante a adaptação do processo foram as seguintes práticas do CMMI: SP 1.1 Obter Entendimento dos Requisitos, SP 1.2 Obter Comprometimento com os Requisitos, SP 1.3 Gerenciar Mudanças nos Requisitos (área de processo Gerenciamento de Requisitos). E para a área de Gerenciamento de Configuração foram: SP 1.1 Identificar Itens de Configuração, SP 1.2 Estabelecer um Sistema de Gestão de Configuração, SP 1.3 Criar ou Liberar *Baselines*, SP 3.1 Estabelecer Registros de Gestão de Configuração, SP 3.2 Executar Auditorias de Configuração.

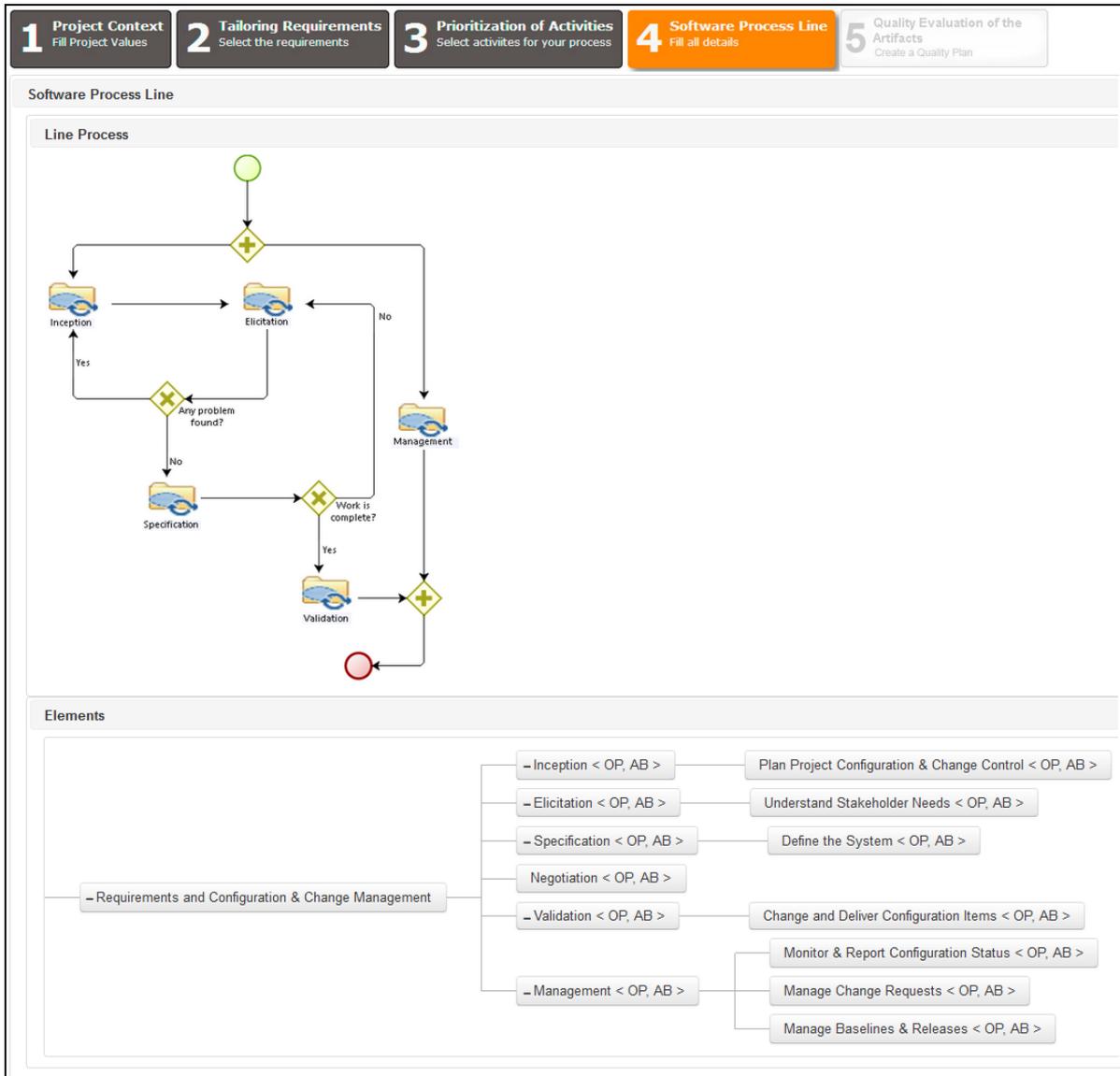
O passo 3 está relacionado à recuperação e priorização das atividades para compor o processo. Sendo assim, algumas atividades foram recuperadas e estas foram priorizadas de acordo com os métodos de priorização baseados em multicritérios (AHP, TODIM e TOPSIS), já citados na Seção 3.2. Para o método AHP quanto maiores os valores de porcentagem, significa que o contexto da atividade é mais similar ao contexto do projeto. Para o TODIM, valores mais próximos de 1 possuem contexto mais adequado ao contexto do projeto, já o método TOPSIS, atividades com maiores valores são mais recomendadas.

A avaliação destes três métodos contribui para a tomada de decisão do engenheiro de processos. Neste caso, após a análise dos resultados dos métodos de priorização, as atividades que foram selecionadas para compor o processo foram: *Plan project configuration & change control, Understand stakeholder needs, Define the system, Change and deliver configuration items, Monitor & report configuration status, Manage change requests, Manage baselines & releases.*

No próximo passo da ferramenta é exibida a LPrS adaptada (Figura 25) para o projeto SIS-ASTROS, contendo as atividades recuperadas, priorizadas e selecionadas para o processo do projeto.

No passo 5, denominado *Quality Evaluation of Artifacts*, é o responsável por gerar os planos de qualidade que correspondem aos artefatos das atividades que compõem a LPrS. Foram gerados planos de qualidade para os seguintes artefatos: Plano de Gerenciamento de Configuração (Tabela 12), Solicitação dos *Stakeholders* (Tabela 25), Especificação de Requisitos de Software (Tabela 24), Especificações Suplementares (Tabela 22), *Workspace* (Tabela 28), Registro de Auditoria de Configuração (Tabela 16), Pedido de Mudança (Tabela 3), Resultados do Teste (Tabela 27), *Test Log* (Tabela 29) e Ordem de Serviço (Tabela 18).

Figura 25 – LPrS do Projeto SIS-ASTROS



Portanto, na etapa 2 deste estudo de caso foi entregue ao participante um documento contendo: a LPrS adaptada para o projeto, as atividades com os seus elementos descritos no Apêndice B e seus respectivos planos de qualidade dos artefatos. Sendo assim, com os dados da segunda etapa do estudo de caso, os participantes puderam analisar o documento e então foi realizada a entrevista com a coordenadora de projetos.

6.1.1 Avaliação do Especialista do Projeto SIS-ASTROS

A coordenadora do projeto SIS-ASTROS explicou que atualmente é utilizado um processo evolutivo, no entanto, este não está bem definido. Algumas diretrizes do processo são as reuniões junto ao cliente, no qual é possível definir os requisitos, depois ocorre a

implementação destes e, posteriormente, é apresentado o produto desenvolvido a partir dos requisitos. Este produto é avaliado junto ao cliente, e com isso podem surgir novos requisitos e o ciclo de desenvolvimento vai seguindo dessa maneira.

A participante afirma que por mais que este processo não esteja bem definido, ele possui características tanto de métodos planejados quanto de métodos ágeis. A equipe de programação utiliza algumas práticas e artefatos do método ágil SCRUM, por exemplo o *Backlog*. Dos processos planejados é utilizado alguns artefatos do RUP, principalmente aqueles documentos que vão ser entregues para o Exército Brasileiro, visto que estes devem ser completos e bem descritos. Portanto, para a especificação de requisitos e gerenciamento de projetos são utilizadas atividades provenientes de processos planejados, no entanto, internamente, a equipe de programação utiliza método ágil.

Os artefatos desenvolvidos pelo projeto SIS-ASTROS até o momento, são: Documento de Doutrina, no qual possui um diagrama de modelagem de negócio para modelar o funcionamento do REOP; Documento Visão, elaborada para ter uma visão global do simulador tático do REOP e Especificação dos Casos de Uso. Além destes artefatos de requisitos, existem artefatos relacionados ao gerenciamento do projeto, tais como: planos de projeto, atas de reunião, solicitações de compras e relatórios de acompanhamento de custos. Também pretende-se trabalhar com o artefato de Especificação de Testes. A coordenadora ressalta a importância de artefatos de documentação bem elaborados, pois como neste projeto o cliente está em Brasília, os artefatos são enviados para o cliente e este envia um *feedback*, portanto já que os contatos presenciais com o cliente não são frequentes, os artefatos auxiliam na discussão a distância.

Em relação aos planos de qualidade, a participante acredita que eles podem ser usados tanto por equipes grandes, quanto por pequenas. O que deve ser alterado são os tipos de métricas adequadas a equipe e ao processo. Se considerar processos planejados, vai ter métricas associadas a documentação, pois esta deve estar compreensível e legível. Enquanto para métodos ágeis, serão métricas mais práticas e que representam o progresso, por exemplo, o artefato *Sprint Burndown* do SCRUM, no qual é acompanhado o progresso diário da equipe, os testes medem o progresso do desenvolvimento, medem a produtividade.

Sobre a possibilidade de alterar os planos, a participante achou relevante, pois provavelmente no início o especialista irá usar os planos conforme está proposto, mas à medida que vai aplicando estes planos, vai adquirindo experiência e isso possibilita que ele os altere para adequá-los a realidade da sua equipe.

A entrevistada avalia que os planos possuem um fácil entendimento e achou bem organizado, visto que está dividido em dados do CMMI, dados do RUP e avaliação dos artefatos. Como sugestão seria adicionar a descrição das medidas, para o melhor entendimento das mesmas.

6.2 ESTUDO DE CASO 2 – PROJETO DA EMPRESA DATAPREV

Neste estudo de caso, foi entrevistado o analista de conformidade da Empresa de Tecnologia e Informações da Previdência (Dataprev), localizada na Unidade de Desenvolvimento de Florianópolis-SC. A Dataprev é uma empresa pública brasileira de grande porte, vinculada ao Ministério da Fazenda. É responsável pela gestão da Base de Dados Sociais Brasileira, especialmente a do Instituto Nacional do Seguro Social (INSS).

Este projeto consiste em monitorar agendamentos da Agência de Previdência Social (APS). Esses agendamentos podem ocorrer tanto por telefone, quanto pessoalmente. Além disso, o sistema deve buscar as informações necessárias ao atendimento do segurado em um sistema central que, por sua vez, acessa o banco de dados do INSS. Atualmente, são vários sistemas usados nas APSs, por exemplo, sistema para buscar os dados dos segurados, sistema para controlar a fila, sistemas de concessão de benefícios, de verificação e controle de fraudes, entre outros, em vista disto, este projeto tem o objetivo de integrar estes sistemas.

O contexto situacional deste projeto foi disponibilizado como mostra a Tabela 6. Para a realização da etapa 2, foi cadastrado o projeto da empresa Dataprev, assim como seu contexto na ferramenta de adaptação de processos.

Tabela 6 – Contexto do Projeto da Empresa Dataprev

Fatores	Atributos
Tamanho da equipe	Pequena (até 12 pessoas)
Arquitetura estável	Estável
Modelo de negócio	Sob medida
Distribuição da equipe	Local
Taxa de mudanças	Mais que 30% no mês
Idade do sistema	Novo desenvolvimento
Criticidade	Perda de dinheiro
Controle	Dinâmico/Flexível

Na entrevista o participante reforça que a equipe é pequena e é composta por um gestor de projeto e outras cinco pessoas que são multifuncionais. O passo que mostra a seleção das práticas do CMMI que deseja satisfazer no processo está na Figura 26. Nesta figura são apresentados os requisitos de adaptação selecionados pelo participante, tanto da

área de processo de Gerenciamento de Requisitos quanto de Gerenciamento de Configuração e as atividades que correspondem a esses requisitos. Vale lembrar que foi selecionada a arquitetura *Requirements and Configuration & Change Management* com as atividades do *framework RUP*.

Figura 26 – Requisitos de Adaptação selecionados para o Projeto da Empresa Dataprev

CMMI - Requirements Management: Level 2	
<input checked="" type="checkbox"/> SP 1.1 Understand Requirements	<input type="checkbox"/> SP 1.2 Obtain Commitment to Requirements
<input checked="" type="checkbox"/> SP 1.3 Manage Requirements Changes	<input checked="" type="checkbox"/> SP 1.4 Maintain Bidirectional Traceability of Requirements
<input type="checkbox"/> SP 1.5 Ensure Alignment Between Project Work and Requirements	
CMMI - Configuration Management: Level 2	
<input type="checkbox"/> SP 1.1 Identify Configuration Items	<input type="checkbox"/> SP 1.2 Establish a Configuration Management System
<input checked="" type="checkbox"/> SP 1.3 Create or Release Baselines	<input checked="" type="checkbox"/> SP 2.1 Track Change Requests
<input type="checkbox"/> SP 2.2 Control Configuration Items	<input type="checkbox"/> SP 3.1 Establish Configuration Management Records
<input type="checkbox"/> SP 3.2 Perform Configuration Audits	
Activities	
<ul style="list-style-type: none"> • Understand Stakeholder Needs • Define the System • Manage the Scope of the System • Manage Changing Requirements • Plan Project Configuration & Change Control • Create Project Configuration Management (CM) Environments • Manage Change Requests • Monitor & Report Configuration Status • Change and Deliver Configuration Items • Manage Baselines & Releases 	

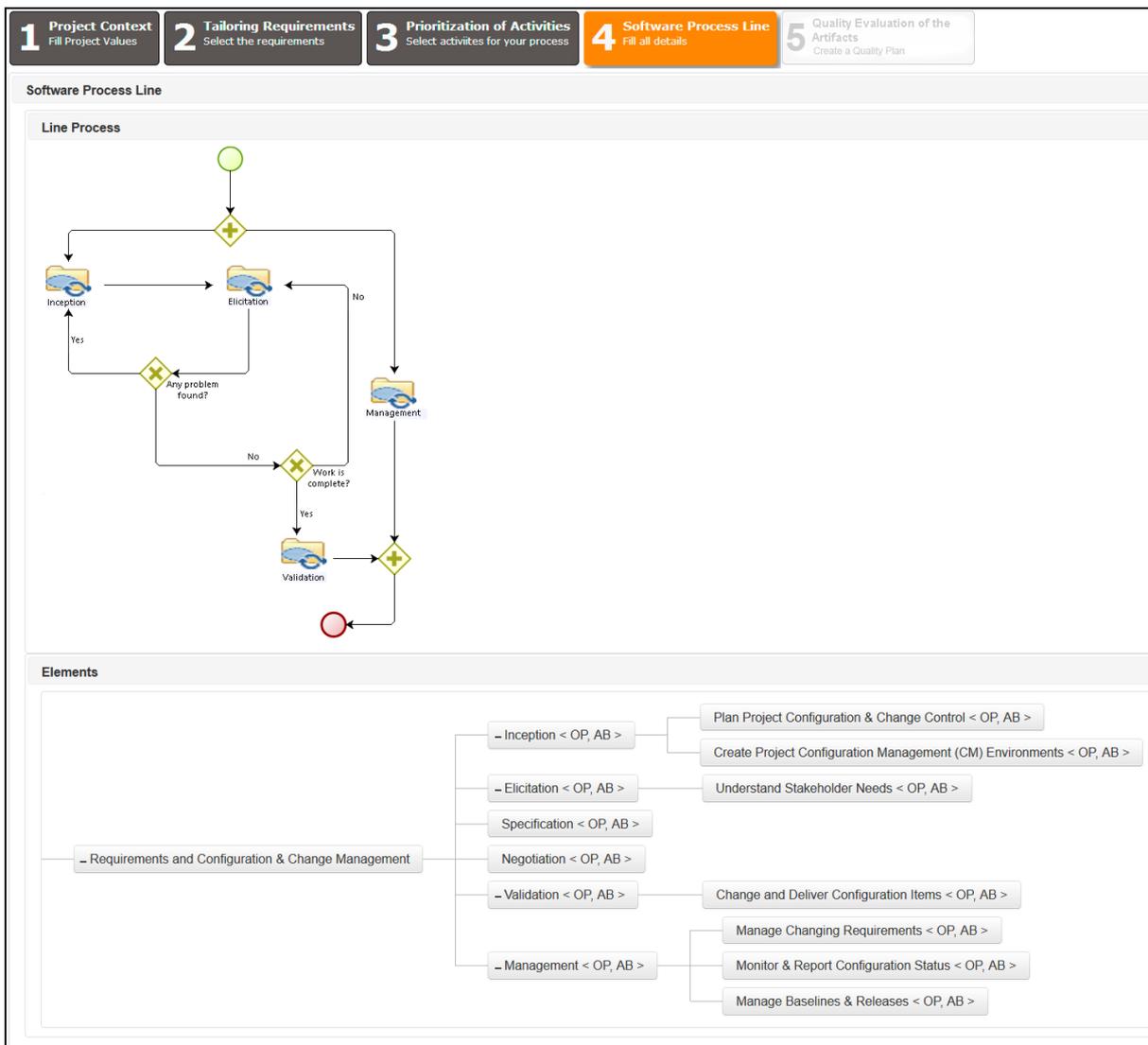
Na Figura 27 são apresentadas todas as atividades recuperadas e quais foram selecionadas, com os seus respectivos resultados de priorização dos três métodos (AHP, TODIM e TOPSIS). Pode-se perceber que as atividades selecionadas para compor o processo adaptado foram aquelas com probabilidades mais altas, pois vale ressaltar que quanto maiores os valores de porcentagem, significa que o contexto da atividade é mais similar ao contexto do projeto.

Figura 27 – Priorização e Seleção das Atividades para o Processo do Projeto Dataprev

1 Project Context Fill Project Values		2 Tailoring Requirements Select the requirements		3 Prioritization of Activities Select activities for your process		4 Software Process Line Fill all details		5 Quality Evaluation of the Artifacts Create a Quality Plan	
Prioritization						Prioritization			
Activities						AHP ↓	TODIM ↓	TOPSIS ↓	
<input checked="" type="checkbox"/>	Understand Stakeholder Needs					11,234 %	0,398	0,551	
<input type="checkbox"/>	Define the System					7,300 %	0,311	0,458	
<input type="checkbox"/>	Manage the Scope of the System					8,949 %	0,000	0,309	
<input checked="" type="checkbox"/>	Manage Changing Requirements					10,115 %	0,217	0,366	
<input checked="" type="checkbox"/>	Plan Project Configuration & Change Control					10,729 %	0,888	0,604	
<input checked="" type="checkbox"/>	Create Project Configuration Management (CM) Environments					10,115 %	0,217	0,366	
<input type="checkbox"/>	Manage Change Requests					9,105 %	0,278	0,436	
<input checked="" type="checkbox"/>	Monitor & Report Configuration Status					12,993 %	1,000	0,691	
<input checked="" type="checkbox"/>	Change and Deliver Configuration Items					10,055 %	0,802	0,604	
<input checked="" type="checkbox"/>	Manage Baselines & Releases					9,407 %	0,322	0,396	

Sendo assim, a LPrS adaptada para o projeto da empresa Dataprev, assim como a arquitetura composta pelos componentes dos processos e suas respectivas atividades são ilustradas na Figura 28.

Figura 28 – LPrS do Projeto da Empresa Dataprev



Os planos de qualidade que correspondem aos artefatos das atividades que compõem a LPrS, foram os seguintes: Plano de Gerenciamento de Configuração (Tabela 12), Repositório do Projeto (Tabela 14), Solicitação dos *Stakeholders* (Tabela 25), *Workspace* (Tabela 28), Registro de Revisão (Tabela 20), Registro de Auditoria de Configuração (Tabela 16), Ordem de Serviço (Tabela 18), encontrados no Apêndice C.

6.2.1 Avaliação do Especialista do Projeto da Empresa Dataprev

Na entrevista o participante explicou que a empresa Dataprev possui dois tipos de processos: um tradicional baseado no RUP e outro ágil baseado nas práticas do SCRUM. O participante afirma que os processos baseados no RUP estão em desuso, visto que exigem muita burocracia.

Este projeto em questão segue o processo com as práticas ágeis. O dono do projeto (*Product Owner*) fica responsável por priorizar os itens (funcionalidades/requisitos) do *Backlog* e à equipe cabe a responsabilidade de se organizar para desenvolver cada item. O processo define um *timebox* (*Sprints*) entre uma e quatro semanas. Neste caso a equipe definiu as *Sprints* de três semanas, ao qual cada item do *Backlog* deve necessariamente estar especificado, codificado e testado manualmente para ser considerado pronto. Caso contrário, volta ao *Backlog*.

Um ponto interessante desta empresa é o fato de que em cada início de um novo projeto, um *checklist* (em torno de 6 perguntas) é usado para verificar se o projeto é aderente às práticas ágeis definidas no processo. Isso é feito por meio de um cálculo da porcentagem das respostas e é analisada a aderência ao processo ágil. Caso o resultado do cálculo seja de pelo menos 75% de aderência, será aplicado o processo ágil, caso contrário, o projeto deve seguir o processo tradicional.

Os artefatos utilizados no projeto da Dataprev são: artefatos de especificação (histórias de usuários, casos de uso, descrição das regras funcionais, requisitos, *layout*), artefatos de codificação (código-fonte) e artefatos de testes (testes unitários, manuais e automatizados).

Em relação a qualidade, esta empresa não utiliza nenhum modelo de qualidade. No entanto, o participante afirma que com um artefato defeituoso há o risco de atrasar o projeto ou de impactar nos custos do mesmo. Ele citou um exemplo no qual a documentação de um determinado projeto era antiga, e assim, a equipe teve problemas de entendimento dos requisitos. A equipe não conseguiu saber realmente o que era para fazer e teve o risco de implementar um requisito errado, ou seja, fazer algo diferente do que o cliente pediu. Portanto, acabou atrasando o projeto.

Em relação a abordagem deste trabalho no qual o especialista seleciona as práticas do CMMI, o participante disse que é vantajoso, pois se no início do projeto já se sabe de antemão quais práticas do CMMI (ou de outro padrão) deverão ser seguidas, fica mais fácil moldar os artefatos para que essas práticas possam ser alcançadas.

Ressalta ainda, que com base em um plano de qualidade pré-definido, é possível seguir o melhor caminho para adotar quais serão os métodos necessários para fazer a avaliação de cada artefato. Uma história de usuário, por exemplo, serve tanto para o entendimento do cliente e da equipe, quanto a um requisito funcional, e também para criar os cenários de casos de testes funcionais. Então, a equipe escreve a história do usuário e submete ao *Product Owner*. Se ele entender, ele informa a equipe e esta faz os cenários e os casos de testes funcionais, depois disso, implementa o código.

Por fim, o participante conclui que os planos ficaram bem divididos e organizados por cada artefato e como cada um deve ser avaliado. Ressalta que é bem possível que ao longo do projeto os planos venham a ser alterados, portanto a possibilidade de alterá-los é benéfica.

6.3 ESTUDO DE CASO 3 – PROJETO DE INOVAÇÃO EM AUTOMAÇÃO AGRÍCOLA

O projeto de Inovação em Automação Agrícola é um projeto de uma empresa confidencial em parceria com o Instituto SENAI de Inovação Soluções Integradas em Metalmeccânica (ISI SIM), localizado em São Leopoldo-RS. Este projeto auxilia na agricultura de precisão, no qual são instalados equipamentos em vários pontos da plantação, estes fazem uma análise clínica e geram resultados. Os resultados são uma análise do solo, a fim de saber se a terra necessita de algum nutriente, ou seja, o objetivo principal é melhorar o plantio. Este participante exerce o cargo de analista de sistemas neste projeto.

A Tabela 7 mostra o contexto do projeto, preenchido no formulário da etapa 1.

Tabela 7 – Contexto do Projeto de Inovação em Automação Agrícola

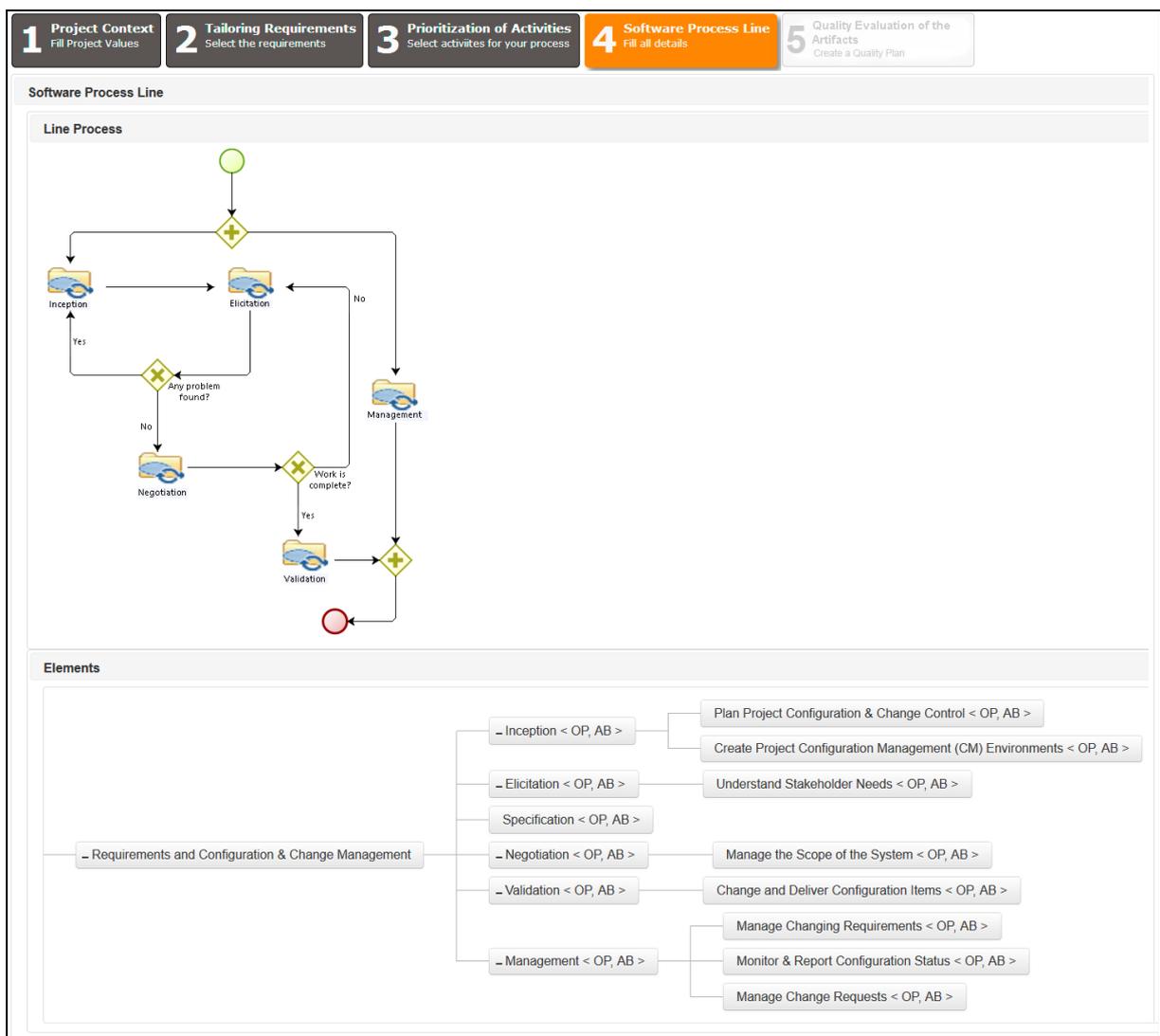
Fatores	Atributos
Tamanho da equipe	Pequena (até 12 pessoas)
Arquitetura estável	Modificada
Modelo de negócio	Comercial
Distribuição da equipe	Equipes diferentes
Taxa de mudanças	Menos que 10% no mês
Idade do sistema	Evolução de sistema legado
Criticidade	Perda de dinheiro
Controle	Regras simples

Dessa forma, tanto o projeto de Inovação em Automação Agrícola quanto o seu contexto foram cadastrados na ferramenta de adaptação de processos. Por sua vez, os requisitos de adaptação selecionados pela participante foram as seguintes práticas do CMMI, para a área de processo Gerenciamento de Requisitos: SP 1.1 Obter Entendimento dos

Requisitos, SP 1.2 Obter Comprometimento com os Requisitos e para a área de Gerenciamento de Configuração foram: SP 1.1 Identificar Itens de Configuração, SP 1.3 Criar ou Liberar *Baselines*, SP 2.1 Acompanhar Solicitações de Mudança, SP 3.2 Executar Auditorias de Configuração.

Na Figura 29 é apresentada a LPrS adaptada para o processo de Inovação em Automação Agrícola, contendo as atividades selecionadas para o processo do projeto em questão.

Figura 29 – LPrS do Projeto de Inovação em Automação Agrícola



Após a recuperação das atividades, foram selecionadas as seguintes atividades: *Plan project configuration & change control*, *Create project configuration management (CM) environments*, *Understand stakeholder needs*, *Manage the scope of the system*, *Change and*

deliver configuration items, Manage changing requirements, Monitor & report configuration status, Manage change requests.

Os planos de qualidade gerados foram para os seguintes artefatos: Plano de Gerenciamento de Configuração (Tabela 12), Repositório do Projeto (Tabela 14), Solicitação dos *Stakeholders* (Tabela 25), Requisitos de Software (Tabela 23), *Workspace* (Tabela 28), Registro de Revisão (Tabela 20), Registro de Auditoria de Configuração (Tabela 16), Pedido de Mudança (Tabela 3), Resultados do Teste (Tabela 27) e *Test Log* (Tabela 29).

6.3.1 Avaliação do Especialista do Projeto de Inovação em Automação Agrícola

Na etapa 3 o participante afirmou que o projeto não possui um processo bem elaborado, apenas foi fornecido uma especificação antiga da empresa, e o cliente afirmou que necessitava de uma melhoria no sistema. O Instituto SENAI, portanto, ficou responsável por essa melhoria com o desenvolvimento *Web* e toda a parte visual e funcional. Ressaltou que por mais que não possuem um processo bem definido, a cada projeto a equipe vai tentando melhorar o seu desenvolvimento. Nesta empresa, já tentaram utilizar o método ágil SCRUM, porém não obtiveram sucesso, e apenas utilizam alguns princípios do mesmo, por exemplo, a comunicação entre os membros da equipe.

Os artefatos que o projeto de Inovação em Automação Agrícola possui são: documento do projeto (descrição inicial do projeto, escopo), planos de negócio (parte inicial do projeto), código-fonte e testes.

O analista de sistemas ressalta que um artefato defeituoso gera atraso no projeto. Um exemplo citado foi o plano de negócio, principalmente no SENAI que é inovação, considera-se bastante processos de compras e de terceirização, então estes precisam estar bem descritos, visto que é um processo muito burocrático. Caso não seja corretamente especificado, eles podem ser bloqueados. Esses processos geralmente duram quatro meses, então se forem feitos ao final do projeto, por exemplo, não vai ter tempo hábil pra termina-lo, gerando um atraso.

Sobre a abordagem desta pesquisa, o analista diz que possui uma relação de selecionar as práticas do CMMI e aumentar a qualidade do processo gerado, principalmente porque já está agregando ao processo uma avaliação da qualidade, isso não vai ser feito depois do processo pronto, vai ser feito durante a fase de adaptação do processo. Com relação aos planos de qualidade gerados, afirma que é possível avaliar um artefato utilizando as informações apresentadas e que em todos os tipos e tamanhos de equipe os planos são importantes. Contudo, em equipes maiores e equipes geograficamente distribuídas os planos

talvez sejam mais importantes, visto que o número de consultas à documentação é maior, e assim, os artefatos devem ser claramente elaborados e descritos.

A possibilidade de alterar os planos de qualidade gerados é importante, pois a equipe tem características específicas, logo precisa de métodos de avaliação adequados a ela. Então, poder alterar/personalizar o plano de qualidade possibilita flexibilizar os planos gerados.

Por fim, o participante conclui afirmando que achou muito interessante a parte de especificar as práticas do CMMI primeiro. Pois é uma forma simples de usar o CMMI. As pessoas geralmente pegam um modelo de qualidade como o CMMI e não sabem como começar, então usar práticas do CMMI para gerar o processo é uma abordagem intuitiva para quem deseja elaborar um processo aderente às suas práticas. E ainda, essa abordagem gera as atividades que vão ser usadas para o processo, e isso faz com que economize esforço e tempo, não precisa analisar o modelo, a abordagem já indica isso.

Uma sugestão do analista seria melhorar a descrição das metas e submetas de qualidade, deixando-as mais claras para o usuário, talvez adicionando uma descrição, porque para o analista de qualidade pode ser simples o entendimento dos planos, no entanto, outro integrante da equipe possa ter dificuldade para compreender as métricas.

6.4 ESTUDO DE CASO 4 – PROJETO PONTO ELETRÔNICO DA UFSM

Este participante trabalha como gerente de projetos no Centro de Processamento de Dados (CPD) da UFSM há mais de 10 anos. O projeto refere-se a substituição do ponto realizado em papel para a implantação de um controle efetivo de frequência de servidores por meio do ponto eletrônico com certificação biométrica. As informações do contexto situacional do projeto foram caracterizadas pelo participante conforme apresentado na Tabela 8.

Tabela 8 – Contexto do Projeto de Ponto Eletrônico da UFSM

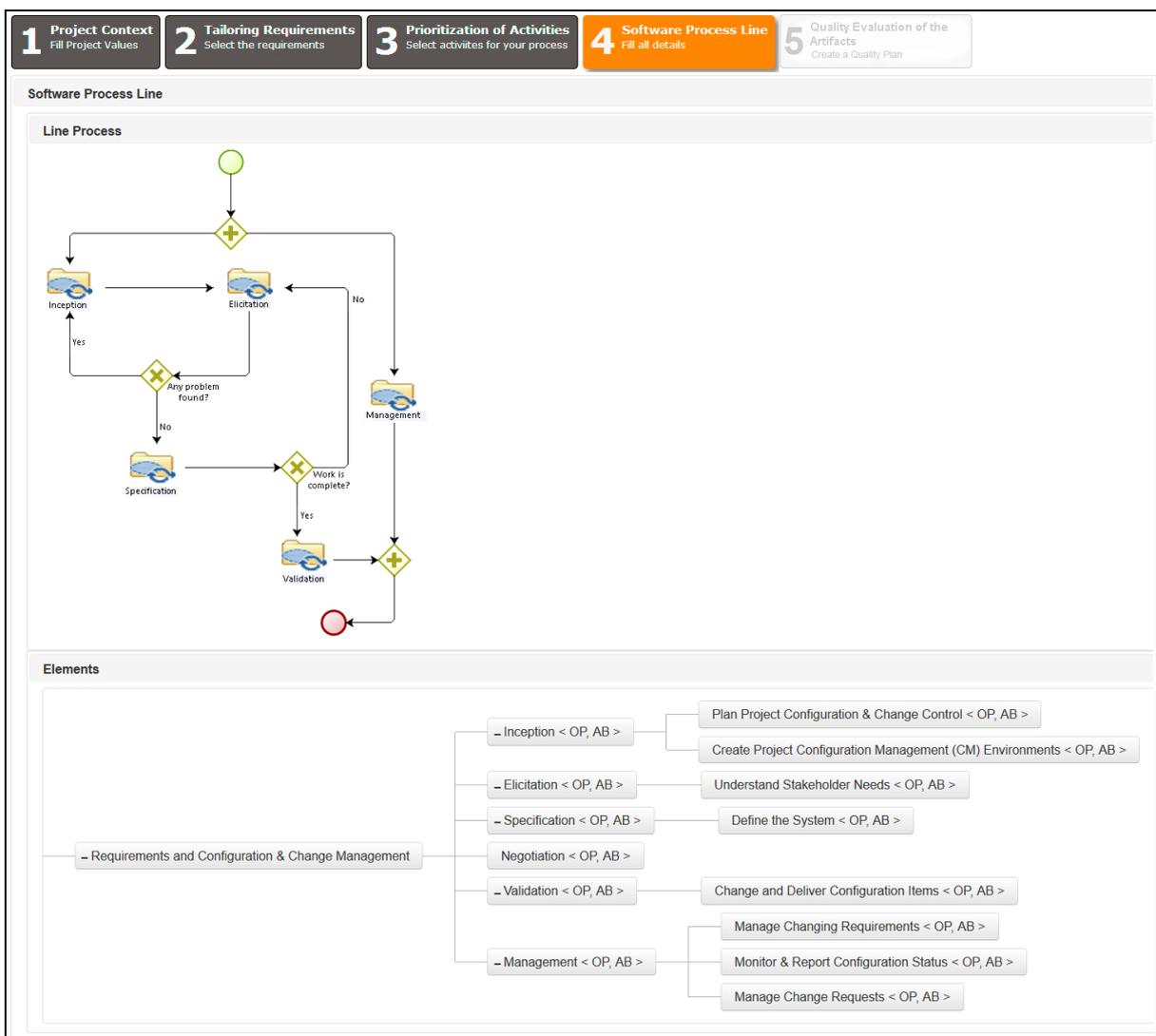
Fatores	Atributos
Tamanho da equipe	Pequena (até 12 pessoas)
Arquitetura estável	Estável
Modelo de negócio	Sob medida
Distribuição da equipe	Local
Taxa de mudanças	Mais que 30% no mês
Idade do sistema	Novo desenvolvimento
Criticidade	Perda de dinheiro
Controle	Dinâmico/Flexível

Do mesmo modo, no formulário da etapa 1, o participante selecionou as práticas do CMMI que deseja satisfazer no processo, que são: para a área de processo Gerenciamento de

Requisitos, SP 1.1 Obter Entendimento dos Requisitos, SP 1.2 Obter Comprometimento com os Requisitos, SP 1.3 Gerenciar Mudanças nos Requisitos e para a área de Gerenciamento de Configuração foram: SP 1.1 Identificar Itens de Configuração, SP 1.2 Estabelecer um Sistema de Gestão de Configuração, SP 1.3 Criar ou Liberar *Baselines*, SP 2.1 Acompanhar Solicitações de Mudança, SP 3.1 Estabelecer Registros de Gestão de Configuração.

Na Figura 30 é exibida a LPrS adaptada contendo as atividades recuperadas, priorizadas e selecionadas para o processo do projeto em questão.

Figura 30 – LPrS do Projeto de Ponto Eletrônico da UFSM



Após a análise dos resultados dos métodos de priorização, as atividades que foram selecionadas para compor o processo foram: *Plan project configuration & change control*, *Create project configuration management (CM) environments*, *Understand stakeholder*

needs, Define the system, Change and deliver configuration items, Manage changing requirements, Monitor & report configuration status, Manage change requests.

Foram gerados planos de qualidade para os seguintes artefatos: Plano de Gerenciamento de Configuração (Tabela 12), Repositório do Projeto (Tabela 14), Solicitação dos *Stakeholders* (Tabela 25), Especificação de Requisitos de Software (Tabela 24), Especificações Suplementares (Tabela 22), *Workspace* (Tabela 28), Registro de Revisão (Tabela 20), Registro de Auditoria de Configuração (Tabela 16), Pedido de Mudança (Tabela 3), Resultados do Teste (Tabela 27) e *Test Log* (Tabela 29).

6.4.1 Avaliação do Especialista do Projeto de Ponto Eletrônico da UFSM

O projeto Ponto Eletrônico é composto por uma equipe pequena de três pessoas, incluindo o gerente do projeto. O entrevistado comentou que a maioria dos projetos possuem equipes pequenas e isto se deve ao fato da alta demanda do CPD. Com isso, gera uma fila grande de projetos e a equipe divide-se em equipes menores com o objetivo de atender o maior número de projetos, visto que existe uma alta pressão externa.

Para este projeto em questão foram definidas algumas atividades para o processo, pois este projeto tem datas de entregas estabelecidas. Normalmente os processos são adaptados conforme a equipe, uma vez que pode ocorrer troca da gestão interna e assim, as visões também serão alteradas.

Em relação aos processos ágeis, houve uma tentativa para aplicação do método SCRUM, no entanto, apenas conseguem utilizar algumas práticas como equipes auto gerenciáveis, no qual as atividades são divididas na equipe e cada um se gerencia dentro das suas responsabilidades. Além disso, o cliente está envolvido no projeto com reuniões periódicas, participando ativamente no processo.

Não utilizam nenhum modelo de qualidade e o controle de qualidade é a satisfação do usuário. Além disso, para este projeto não foram utilizados artefatos, porém o gerente afirma que a equipe tem falhado em relação a isso. O participante acredita que os artefatos são importantes, mas devido à pressão externa e por considerar que a documentação gasta um tempo significativo, a elaboração de artefatos acaba não ocorrendo. Todavia, ressalta a importância dos mesmos, principalmente caso ocorra modificação na equipe, ter uma documentação bem elaborada iria facilitar muito.

Os planos de avaliação da qualidade estão de acordo com a proposta do trabalho, no entanto, o participante acredita que devem ser mais usados em equipes maiores e com

bastante artefatos. A possibilidade de alteração dos mesmos é importante, pois não deixa o plano de qualidade engessado com aqueles determinados métodos de avaliação e métricas.

6.5 ESTUDO DE CASO 5 – PROJETO SISTEMA IRRIGA

Este estudo de caso foi realizado com o gerente de projetos do Sistema Irriga. O Sistema Irriga gerencia o manejo e monitoramento de irrigações a serem aplicadas pelos diferentes métodos de irrigação. As recomendações de quando irrigar e a quantidade de água que deve ser aplicada em cada irrigação estão disponibilizadas por meio de acesso ao portal de internet do Sistema Irriga, que disponibiliza as recomendações 24 horas por dia, sete dias por semana. A vantagem deste sistema é que o produtor reduz o risco de perdas de produtividade devido ao estresse ou excesso hídrico, além de aumentar a produtividade das culturas e economizar água e energia.

O contexto com os fatores do *Octopus Model*, preenchido na etapa 1, relacionado a este projeto encontra-se na Tabela 9.

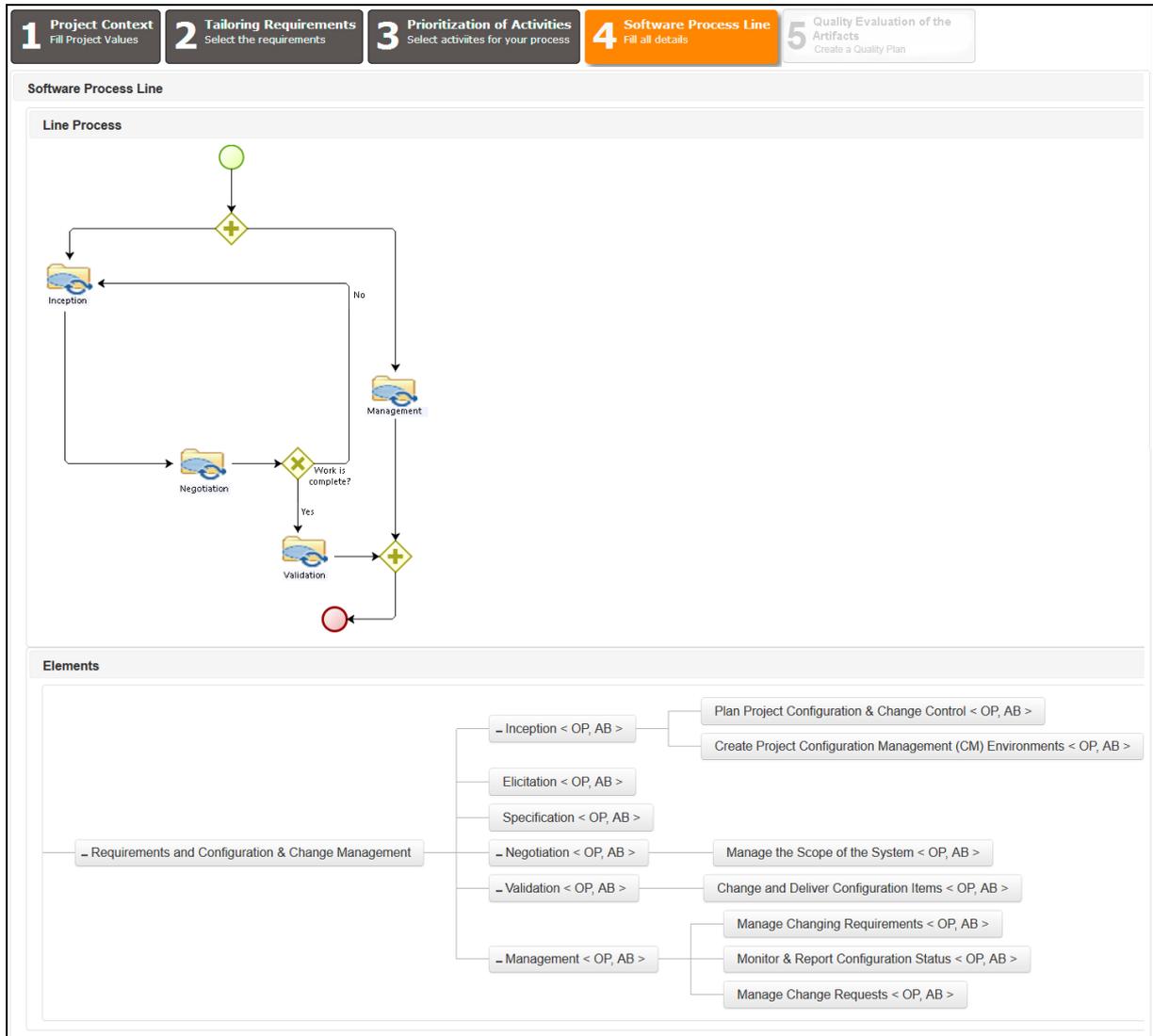
Tabela 9 – Contexto do Projeto Sistema Irriga

Fatores	Atributos
Tamanho da equipe	Pequena (até 12 pessoas)
Arquitetura estável	Modificada
Modelo de negócio	Comercial
Distribuição da equipe	Distribuição geográfica
Taxa de mudanças	Mais que 30% no mês
Idade do sistema	Evolução de sistema legado
Criticidade	Perda de dinheiro
Controle	Regras simples

Por sua vez, os requisitos de adaptação selecionados pelo participante foram as seguintes práticas do CMMI: para a área de processo Gerenciamento de Requisitos, SP 1.1 Obter Entendimento dos Requisitos, SP 1.2 Obter Comprometimento com os Requisitos, SP 1.3 Gerenciar Mudanças nos Requisitos, SP 1.5 Identificar Inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos e para a área de Gerenciamento de Configuração foram: SP 1.1 Identificar Itens de Configuração, SP 1.2 Estabelecer um Sistema de Gestão de Configuração, SP 2.1 Acompanhar Solicitações de Mudança, SP 3.1 Estabelecer Registros de Gestão de Configuração.

Na Figura 31 são apresentadas a LPrS adaptada para o projeto Sistema Irriga, assim como a arquitetura composta pelos componentes dos processos e suas respectivas atividades que foram selecionadas para compor o processo.

Figura 31 – LPrS do Projeto Sistema Irriga



Os planos de qualidade que correspondem aos artefatos das atividades que compõem a LPrS, foram os seguintes: Plano de Gerenciamento de Configuração (Tabela 12), Repositório do Projeto (Tabela 14), Requisitos de Software (Tabela 23), *Workspace* (Tabela 28), Registro de Revisão (Tabela 20), Registro de Auditoria de Configuração (Tabela 16), Pedido de Mudança (Tabela 3), Resultados do Teste (Tabela 27) e *Test Log* (Tabela 29).

6.5.1 Avaliação do Especialista do Projeto Sistema Irriga

O gerente do projeto Sistema Irriga afirma que, atualmente, não possui nenhum processo bem definido, sendo que podem chegar várias demandas ao mesmo tempo na empresa, então o gerente é responsável por analisar as tarefas e efetuar a delegação das

mesmas. No entanto, o participante garantiu que já está em andamento a elaboração de um processo com atividades e artefatos bem definidos. Esta necessidade surgiu, principalmente, pela grande ocorrência de mudanças de requisitos, sendo assim o gerente de projetos pretende utilizar o processo planejado RUP, não em sua totalidade, mais especificamente em artefatos relacionados à disciplina de gerenciamento de requisitos.

A equipe do projeto é pequena, composta por um gerente de projetos e cinco desenvolvedores e analistas de sistemas. Esta equipe utiliza algumas práticas do método ágil SCRUM, como as reuniões diárias e equipe auto gerenciável, de forma que o próprio desenvolvedor define o que vai fazer e administra seu ritmo de trabalho. Além disso, utilizaram o *Kanban* para aumentar a organização e produtividade nas suas atividades diárias, no entanto, o tempo de utilização foi curto. Em outro período utilizaram *Sprint* equivalente a quatro semanas, porém não foi bem sucedida, visto que muitas tarefas foram adicionadas no decorrer da mesma.

Sobre a utilização de algum modelo de qualidade, o participante afirmou que já pensou em usar, contudo ressalta que é muito difícil adequar o modelo conforme a realidade do projeto. Portanto, o entrevistado afirmou que a abordagem criada neste trabalho, no qual as atividades do processo são as que satisfazem as práticas do modelo de qualidade CMMI é vantajoso pelo fato de guiar a equipe em relação as atividades a serem realizadas.

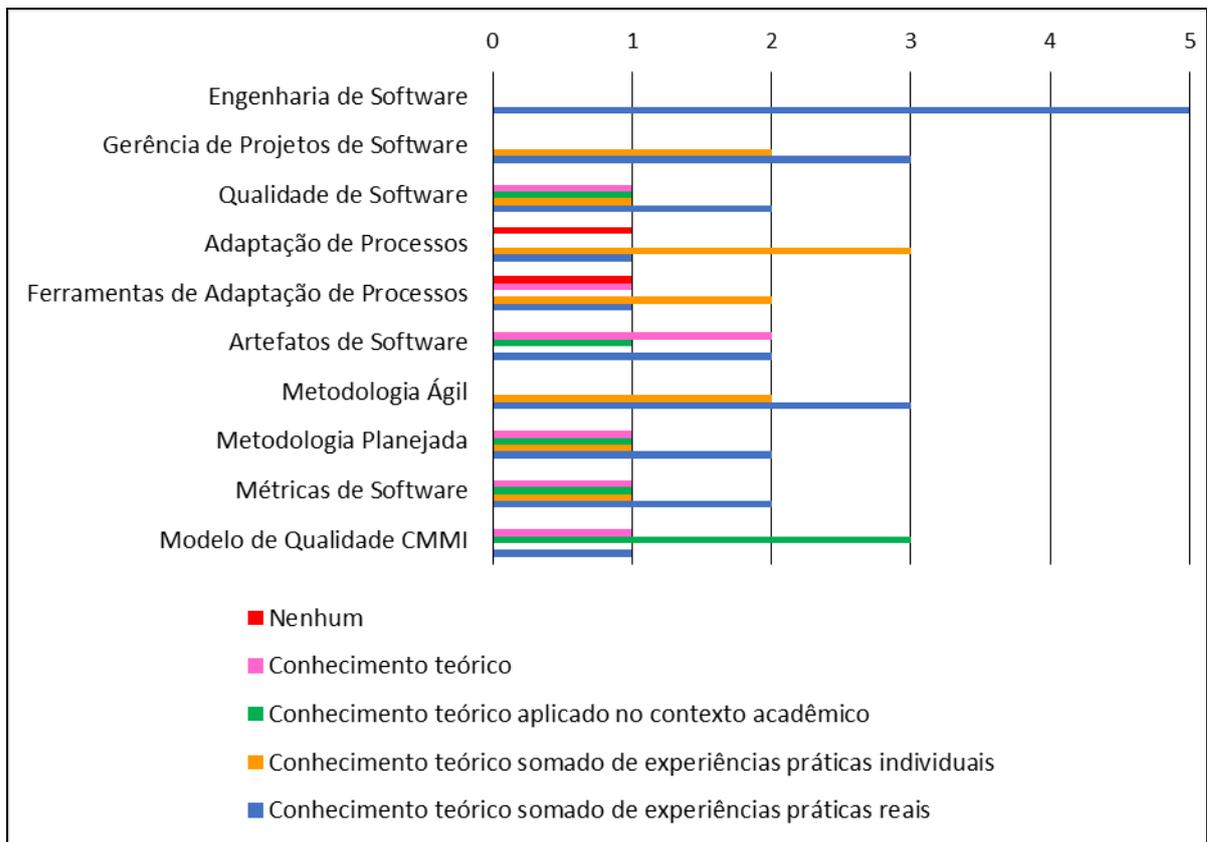
6.6 ANÁLISE DOS ESTUDOS DE CASO

Esta seção resume os principais resultados obtidos com a execução dos estudos de caso. Os estudos de caso foram realizados com cinco especialistas, três destes ocupam o cargo de gerentes de projetos, um analista de conformidade e um analista de sistemas. Cada participante, no entanto, foi solicitado a preencher um formulário de caracterização do seu projeto de software, assim como selecionar as práticas do CMMI que gostaria de satisfazer no processo. Deste modo, projetos com diferentes características (contextos) e práticas selecionadas, resultaram em diferentes processos adaptados.

Na etapa 1 foi coletado por meio do formulário de caracterização do participante o seu grau de experiência e conhecimento quanto aos seguintes assuntos (eixo vertical): Engenharia de Software, Gerência de Projetos de Software, Qualidade de Software, Adaptação de Processos, Ferramentas de Adaptação de Processos, Artefatos de Software, Metodologia Ágil, Metodologia Planejada, Métricas de Software e Modelo de Qualidade CMMI. Pode-se perceber, conforme mostra a Figura 32, que os cinco participantes (eixo horizontal) possuem

alto grau de experiência em Engenharia de Software, Gerência de Projetos de Software e Metodologia Ágil e apenas um participante selecionou que não possui nenhuma experiência com Adaptação de Processos e Ferramentas de Adaptação de Processos.

Figura 32 – Grau de Experiência dos Especialistas



De forma a facilitar o estudo, foi elaborado um documento contendo a LPrS adaptada para cada projeto, as atividades com os seus elementos e seus respectivos planos de qualidade dos artefatos gerados a partir dos dados preenchidos no formulário da etapa 1. O participante podia consultar este material durante a realização da etapa 3 e serviu para auxiliá-lo na análise, posteriormente, então, realizou-se a entrevista.

A primeira parte da entrevista foi relacionada a processos de desenvolvimento de software, no qual é questionado alguns aspectos, tais como: a presença de um processo bem definido dentro da empresa, a utilização de adaptação de processos, o uso de processo ágil e/ou processo planejado e artefatos de software.

Nesta etapa foi constatado que dos cinco projetos, apenas o projeto da empresa Dataprev possui processo de desenvolvimento com atividades e artefatos bem definidos, além de se preocupar com o contexto do projeto, pois utilizam um *checklist* para caracterizá-lo.

Esse *checklist* é usado para decidir se a equipe deve aderir ao processo ágil ou processo planejado. Os demais projetos possuem processos *ad hoc*, no qual não estão totalmente documentados.

Em relação a segunda parte da entrevista é referente a qualidade de software, pretendia saber se a organização utiliza algum modelo de qualidade, se um artefato defeituoso pode atrasar o projeto ou aumentar o custo dele e se selecionando as práticas do CMMI, como a abordagem propõe, aumenta a qualidade do processo. As respostas sobre a utilização de algum modelo de qualidade foram unânimes, nenhum dos projetos usam e ressaltam que é muito difícil adequar o modelo conforme a realidade do projeto. Sobre os artefatos defeituosos, alguns participantes citaram casos da própria organização que exemplificam e que comprovam que podem atrasar o projeto ou o custo dele.

Quanto ao especialista selecionar as práticas do CMMI, os participantes disseram que é vantajoso, pois se no início do projeto já se sabe de antemão quais práticas do CMMI deverão ser seguidas, fica mais fácil moldar os artefatos para que essas práticas possam ser alcançadas. Ressaltam que é uma forma simples de usar o CMMI e que é uma abordagem intuitiva pra quem vai montar o processo. Além de gerar as atividades que vão ser usadas para o processo, e isso faz com que economize esforço e tempo, visto que não precisa analisar o modelo, a abordagem já indicaria isso.

O terceiro tópico levou em consideração a abordagem proposta, com o objetivo de saber a importância da avaliação da qualidade de artefatos de software, em quais equipes a aplicabilidade dos planos de qualidade são mais eficientes, assim como se ficaram satisfeitos com o formato dos planos de qualidade elaborados para a avaliação dos artefatos. Os resultados obtidos com a análise desta questão permitem concluir que os planos propostos podem ser usados tanto por equipes grandes, quanto por pequenas. O que deve ser alterado são os tipos de métricas adequadas a equipe e ao processo. Contudo, um participante acredita que em equipes maiores e equipes geograficamente distribuídas os planos talvez sejam mais importantes, visto que o número de consultas à documentação é maior, e assim, os artefatos devem ser bem elaborados e descritos.

Todos os especialistas afirmaram que é possível avaliar um artefato utilizando as informações apresentadas no plano de avaliação de qualidade proposto. Avaliam que os planos possuem um fácil entendimento e são organizados, visto que estão divididos em dados do CMMI, dados do RUP e avaliação dos artefatos. Ressaltam ainda, que com base em um plano de qualidade pré-definido, é possível seguir o melhor caminho para adotar os métodos necessários para fazer a avaliação de cada artefato.

Em resumo, os participantes deste estudo de caso deram um *feedback* positivo sobre a abordagem proposta, consideraram a ideia de avaliar os artefatos de software gerados ao longo do desenvolvimento de software válida e relevante. No entanto, os participantes identificaram oportunidades de melhorias futuras para o processo de avaliação dos artefatos e elaboração dos planos de qualidade. Foram propostas as sugestões de acrescentar na parte de avaliação dos artefatos uma descrição das medidas, para o melhor entendimento das mesmas e também descrição das metas e submetas de qualidade, para ficar mais claro o real significado desses elementos e melhorar a compreensão dos planos por toda a equipe de desenvolvimento.

7 CONSIDERAÇÕES FINAIS

Produzir software de qualidade é uma meta essencial e básica da Engenharia de Software. Em outras palavras, entregar um produto de software sem falhas garante a confiabilidade, eficiência e integridade do produto. Em vista disso, neste trabalho defende-se a ideia de que a avaliação da qualidade de um produto de software pode ser iniciada já nos primeiros estágios de um processo de desenvolvimento com o objetivo de detectar e corrigir os problemas encontrados antes que estes se propaguem. A preocupação é garantir a qualidade em cada artefato gerado pelas atividades de um processo de desenvolvimento, de modo a produzir artefatos sem defeitos e, conseqüentemente, aumentar a qualidade no processo.

Este trabalho apresentou um *framework* para avaliação da qualidade de artefatos, no qual possui uma LPrS adaptada a partir da reutilização de atividades devidamente caracterizadas. Para definir processos de software de maneira mais efetiva é possível investir na reutilização de processos a partir de processos definidos previamente e na adequação ao contexto do projeto. Neste cenário de reutilização, a ferramenta elabora os processos baseados em estruturas como componentes e linhas de processos. Sendo assim, a ferramenta gera uma LPrS adaptada que utiliza uma arquitetura de processos para os quais são recuperadas e priorizadas atividades para compor essa linha.

O *framework* proposto é composto por um metamodelo, uma base de conhecimento elaborada a partir do modelo de qualidade CMMI, um processo de avaliação, cujo objetivo é organizar os conceitos que envolvem a definição de metas de qualidade e seus respectivos métodos e métricas de avaliação, além de uma ferramenta de apoio. O plano para avaliação de artefatos é elaborado considerando o conjunto de artefatos selecionados durante o processo adaptado e a partir do reuso de instâncias do metamodelo de qualidade.

A validação da abordagem proposta foi realizada por meio de estudos de caso. Estes foram feitos em três etapas, no qual a primeira foi o preenchimento de um formulário de caracterização do perfil do participante e do projeto no qual está inserido. A segunda etapa foi a criação do processo adaptado e dos planos de qualidade e a terceira teve como objetivo entrevistar especialistas de empresas distintas, cujos tópicos da entrevista foram relacionados a processos de software, qualidade de software e planos de qualidade de artefatos.

Em cada um dos projetos dos estudos de casos foi gerado um processo adaptado levando em conta o requisito de adaptação proposto (práticas do modelo de qualidade

CMMI), somado ao contexto do projeto definido pelo *Octopus Model*. Os planos de avaliação de qualidade dos artefatos foram analisados pelos especialistas e os resultados obtidos foram positivos, os participantes concordaram que a abordagem proposta de avaliar os artefatos de software gerados ao longo do desenvolvimento de software é válida e relevante.

Por fim, conclui-se que, ainda que existam pontos a serem melhorados, estudados e implementados em trabalhos futuros, acredita-se que a proposta deste trabalho atingiu seus objetivos. Diante da dificuldade de obtenção de qualidade no processo de desenvolvimento de software, a ideia de avaliar os artefatos que compõem o processo foi uma abordagem interessante e que o controle da qualidade durante o processo de desenvolvimento do produto diminui o retrabalho.

7.1 CONTRIBUIÇÕES

Esta dissertação apresenta, como principal contribuição a proposta de uma abordagem sistemática para a avaliação da qualidade de artefatos gerados e/ou transformados pelas atividades que compõem um processo de software adaptado. Entretanto, outras contribuições do trabalho incluem:

- A proposta de um *framework* de qualidade, composto por um metamodelo, uma base de conhecimento a partir do CMMI, um processo de avaliação e o *QAPro System*;
- A elaboração de um metamodelo, cujo objetivo é organizar os conceitos que envolvem a definição de conceitos de elementos de processos e elementos relacionados a qualidade;
- O relacionamento entre as práticas do modelo de qualidade CMMI com os artefatos e as atividades do RUP;
- A busca por métodos de avaliação e métricas de metas e submetas de qualidade;
- A produção de planos de qualidade a fim de identificar métodos e métricas mais apropriadas para avaliar os artefatos do processo de software adaptado;
- O desenvolvimento do *QAPro System* a partir de uma extensão da ferramenta MfTPt, com a geração de planos de qualidade com vistas a auxiliar os analistas de qualidade na avaliação dos artefatos elaborados ao longo de um processo de software.

7.2 TRABALHOS FUTUROS

Durante a elaboração desta dissertação, alguns estudos futuros foram identificados como oportunidades para complementar esta pesquisa explorando limitações e/ou aspectos que não puderam ser investigados neste trabalho.

Como melhoria para trabalhos futuros, pretende-se realizar uma validação prática da abordagem, ou seja, aplicar os planos de qualidade de artefatos gerados em projetos de desenvolvimento de software reais em diferentes organizações. Com isso, espera-se que seja possível obter um *feedback* sobre os planos de qualidade elaborados e assim, aprimorar e refinar o *framework* proposto a partir das novas opiniões.

Além disso, pretende-se estender a abordagem para o restante das áreas de processo do CMMI, portanto é necessário associar as atividades do RUP com cada prática descrita no modelo. No entanto, visando expandir uma das limitações associadas à abordagem descrita neste trabalho que está no fato de utilizar as atividades do processo planejado RUP, como sugestão seria associar as práticas do CMMI com outras atividades, por exemplo, de métodos ágeis.

Paralelamente, pretende-se elaborar planos de qualidade com métricas relacionadas aos métodos ágeis, por exemplo com medidas de progresso e produtividade, pois as métricas utilizadas nesta dissertação estão mais associadas a documentação em si.

Por fim, outra perspectiva futura é estender a abrangência deste trabalho para os outros requisitos de adaptação já existentes na ferramenta, ou seja, realizar associações dos critérios de qualidade, riscos e segurança.

7.3 PUBLICAÇÕES

A abordagem proposta por esta dissertação foi aceita para publicação no *27th International Conference on Software Engineering and Knowledge Engineering (SEKE 2015)*, intitulado *Quality Evaluation of Artifacts in Tailored Software Process Lines*. Este artigo tem co-autoria da professora Lisandra M. Fontoura e Gelson Bertuol, egresso 2014 do Programa de Pós Graduação em Informática da UFSM.

- BRONDANI, C. H.; BERTUOL, G.; FONTOURA, L. M. Quality Evaluation of Artifacts in Tailored Software Process Lines. XXVII International Conference on Software Engineering and Knowledge Engineering, p. 223–226, 2015.

REFERÊNCIAS

- ALEIXO, F. A. et al. Uma Abordagem para Gerência e Customização de Variabilidades em Processos de Software. **Simpósio Brasileiro de Engenharia de Software (SBES) - Congresso Brasileiro de Software: Teoria e Prática (CBSOFT)**, p. 119–128, 2010.
- AL-KILIDAR, H.; COX, K.; KITCHENHAM, B. The Use and Usefulness of the ISO/IEC 9126 Quality Standard. **International Symposium on Empirical Software Engineering**, p. 126–132, 2005.
- ARMBRUST, O. et al. **Scoping Software Process Models - Initial Concepts and Experience from Defining Space Standards**. Making Globally Distributed Software Development a Success Story. Springer Berlin Heidelberg, v. 5007, p. 160–172, 2008.
- BARRETO, A. S.; MURTA, L. G. P.; ROCHA, A. R. C. Software Process Definition: A Reuse-Based Approach. **Journal of Universal Computer Science**, v. 17, n. 13, p. 1765–1799, 2011.
- BARTIÉ, A. **Garantia da qualidade de software**. Rio de Janeiro: Elsevier, 2002.
- BASILI, V. R.; CALDEIRA, G.; ROMBACH, H. D. The Goal Question Metric Approach. **Encyclopedia of Software Engineering**, v. 2, p. 1–10, 1994.
- BECK, K. **Programação Extrema (XP) Explicada: Acolha as Mudanças**. Bookman, 2004.
- BERNABÉ, O. et al. Building the pillars for the definition of a data quality model to be applied to the artifacts used in the Planning Process of a Software Development Project. **6th Iberian Conference on Information Systems and Technologies (CISTI)**, 2011.
- BERTUOL, G. **Uma Abordagem para Avaliação da Qualidade de Artefatos de Software**. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Maria, Santa Maria, RS, p. 106, 2014.
- BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. 3. ed. Elsevier, 2015.
- BOEHM, B. et al. Characteristics of software quality. **TRW series of software technology**, 1978.
- BOEHM, B.; BROWN, J. R.; LIPOW, M. Quantitative Evaluation of Software Quality. **Proceedings of the 2nd International Conference on Software Engineering**, p. 592–605, 1976.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Unified Modeling Language Users**

Guide. 2. ed. Addison Wesley, 2005.

BRASIL, M. A. B.; FONTOURA, L. M.; ALVARO, L. Uma Proposta para Melhoria da Qualidade de Processos de Software com base em MPS-BR. **XII Simpósio Brasileiro de Qualidade de Software (SBQS)**, p. 123–137, 2013.

BRONDANI, C. H.; BERTUOL, G.; FONTOURA, L. M. Quality Evaluation of Artifacts in Tailored Software Process Lines. **XXVII International Conference on Software Engineering and Knowledge Engineering**, p. 223–226, 2015.

CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. **CMMI: Guidelines for Process Integration and Product Improvement**. 2. ed. Boston, USA: Addison-Wesley, 2006.

COPLIEN, J. O.; ALEXANDER, A. W. O. Software Patterns. **Citeseer**, p. 1–29, 1996.

DEISSENBOECK, F. et al. Software quality models: purposes, usage scenarios and requirements. **7th International Workshop on Software Quality**, p. 9–14, 2009.

ERICSSON, E. et al. Process Improvement Framework Evaluation. **International Conference on Management Science and Engineering (ICMSE)**, p. 319–326, 2010.

FONTOURA, L. M. **PRiMA: Project Risk Management Approach**. Tese (Doutorado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, p. 206, 2006.

FUGGETTA, A. Software Process: A Roadmap. **22nd International Conference on Software Engineering**, 2000.

GUERRA, A. C.; COLOMBO, R. M. T. **Tecnologia da Informação: Qualidade de Produto de Software**. Brasília: PBQP Software, 2009.

IBM RATIONAL. **Rational Unified Process: Version 7.2**. 2003.

ISO/IEC 12207. **Systems and software engineering - Software life cycle processes**. 2008.

ISO/IEC 14598. **Information Technology - Software Product Evaluation**. 1999.

ISO/IEC 15504. **Information Technology. Process Assessment. Part 1: Concepts and Vocabulary**. 2003.

ISO/IEC 25000. **Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE**. 2005.

ISO/IEC 9000. **Sistemas de Gestão da Qualidade - Fundamentos e Vocabulário**. 2005.

ISO/IEC 9126. **Software Engineering - Product Quality**. 2003.

JAKOBSEN, C. R.; JOHNSON, K. A. Mature Agile with a twist of CMMI. **Conference Agile**, p. 212 – 217, 2008.

JAUFMAN, O.; MÜNCH, J. Acquisition of a project-specific process. **Product Focused Software Process Improvement**, v. 3547, p. 328–342, 2005.

JING, L. Application of CMMI in Innovation Management. **International Conference on Wireless Communications, Networking and Mobile Computing**, p. 4966–4969, 2007.

KREBS, J.; SHUJA, A. K. IBM Rational Unified Process Reference and Certification Guide. **Solution Design**. Prentice Hall, 2008.

KRUCHTEN, P. Contextualizing agile software development. **Journal of Software: Evolution and Process**, v. 25, n. 4, p. 351–361, 2010.

LEE, K.; LEE, S. J. A quantitative software quality evaluation model for the artifacts of component based development. **Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05)**, 2005.

LEE, S.; SHIM, J.; WU, C. A metal model approach using uml for task assignment policy in software process. **Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC)**, 2002.

LORENZ, W. G. **Uma sistemática baseada no reuso de atividades para adaptação de processos de software utilizando linhas de processos de software**. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Maria, Santa Maria, RS, p. 94, 2014.

LORENZ, W. G. et al. Activity-based software process lines tailoring. **International Journal of Software Engineering and Knowledge Engineering**, v. 24, n. 09, 2014.

MAGDALENO, A. M. et al. Flexible Organizational Process Deployment. **Computer Supported Cooperative Work in Design III**, v. 4402, p. 679– 688, 2007.

MAGDALENO, A. M. An optimization-based approach to software development process tailoring. **2nd International Symposium on Search Based Software Engineering**, p. 40–43, 2010.

MCCALL, J. A.; RICHARDS, P. K.; WALTERS, G. F. Factors in Software Quality. **Nat'l Tech. Information Servicel**, v. 1, 2, 3, 1977.

MELLOR, S. et al. **MDA Distilled - Principles of Model Driven Architecture**. Addison-Wesley, 2004.

MOHAGHEGHI, P.; DEHLEN, V.; NEPLE, T. Towards a tool-supported quality model for model-driven engineering. **Proc. 3rd International Workshop on Quality in Modeling**, 2008.

NORTHROP, L. M. SEI's software product line tenets. **IEEE Software**, v. 19, n. 4, p. 32–40, 2002.

NUNES, E. et al. Definição de Processos de Aquisição de Software para Reutilização. **Conferência Latino-americana de Informática (CLEI)**, 2010.

PEDREIRA, O. et al. A systematic review of software process tailoring. **SIGSOFT Software Engineering Notes**, v. 32, n. 3, p. 1–6, 2007.

PEREIRA, G. V.; SEVERO, F.; FONTOURA, L. M. A MultiCriteria Risk Prevention Approach. **IADIS International Conference Applied Computing**, 2012.

PERSCH, H. M. **Elaboração de Processos Seguros por meio da Priorização de Componentes de Processo Reusáveis definidos a partir de Padrões de Segurança**. Trabalho de Graduação (Bacharelado em Sistemas de Informação) - Universidade Federal de Santa Maria, Santa Maria, RS, p. 48, 2014.

PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. 7. ed. McGraw-Hill, 2011.

PROJECT MANAGEMENT INSTITUTE - PMI. **A Guide to the Project Management Body of Knowledge**. 5. ed. Pennsylvania: Project Management Institute, Inc., 2013.

RODRÍGUEZ, M. et al. A methodology for continuous quality assessment of software artifacts. **10th International Conference on Quality Software**, p. 254–261, 2010.

ROMBACH, D. Integrated software process and product lines. **International Conference on Unifying the Software Process Spectrum**, p. 83–90, 2006.

SCHWABER, K.; BEEDLE, M. **Agile Software Development with Scrum**. 1. ed. Prentice Hall, 2001.

SEI. **CMMI® for Development, Version 1.3**. 2010.

SOFTEX. **MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral**. 2011.

SOLINGEN, R. VAN. Measuring the ROI of Software Process Improvement. **IEEE Computer Society**, v. 21, p. 32–38, 2004.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

STORCH, A.; LAUE, R.; GRUHN, V. Measuring and visualising the quality of models. **1st International Workshop on Communicating Business Process and Software Models Quality, Understandability, and Maintainability (CPSM)**, 2013.

SURYN, W.; ABRAN, A. ISO/IEC SQuaRE: The Second Generation of Standards for Software Product Quality. **International Association of Science and Technology for Development - IASTED**, p. 1–11, 2003.

TERNITÉ, T. Process Lines: A Product Line Approach Designed for Process Model Development. **35th Euromicro Conference on Software Engineering and Advanced Applications**, p. 173–180, 2009.

TRENDOWICZ, A.; PUNTER, T. Quality modelling for software product lines. **7th Workshop on Quantitative Approach in Object-Oriented Software Engineering**, 2003.

VANDEROSE, B. **Supporting a model-driven and iterative quality assessment methodology: The MoCQA framework**. Tese (Doutorado em Ciência da Computação) - PReCISE Research Centre Faculty of Computer Science University of Namur (FUNDP) Namur, Belgium, p. 314, 2012.

VANDEROSE, B.; HABRA, N. Tool-support for a model-centric quality assessment: QuaTALOG. **Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement**, p. 263–268, 2011.

WASHIZAKI, H. Building software process line architectures from bottom up. **Product-Focused Software Process Improvement**, v. 4034, p. 415–421, 2006a.

WASHIZAKI, H. Deriving Project-Specific Processes from Process Line Architecture with Commonality and Variability. **4th IEEE International Conference on Industrial Informatics**, p. 1301–1306, 2006b.

XU, P.; RAMESH, B. Software Process Tailoring: An Empirical Investigation. **Journal of Management Information Systems**, v. 24, n. 2, p. 293–328, 2007.

XU, P.; RAMESH, B. Using Process Tailoring to Manage Software Development Challenges. **IEEE Computer Society**, p. 39–45, 2008.

APÊNDICES

APÊNDICE A – ASSOCIAÇÃO ENTRE CMMI E RUP

Tabela 10 – Associação da Área de Processo Gerenciamento de Requisitos

CMMI: Level 2 - Process Area: Requirements Management (REQM)			
Specific Goals CMMI	Specific Practices CMMI	Artifacts RUP	Activities RUP
SG 1 Manage Requirements	SP 1.1 Understand Requirements	Requirements Attributes (R)	Analyze the Problem (R)
		Supplementary Specifications (R)	Understand Stakeholder Needs (R)
		Software Requirement (R)	Define the System (R)
		Software Requirements Specification (R)	Manage the Scope of the System (R)
		Requirements Management Plan (R)	Refine the System Definition (R)
	SP 1.2 Obtain Commitment to Requirements	Stakeholder Requests (R)	Manage Changing Requirements (R)
		Change Request (C)	Analyze the Problem (R)
			Understand Stakeholder Needs (R)
			Define the System (R)
			Manage the Scope of the System (R)
	SP 1.3 Manage Requirements Changes	Stakeholder Requests (R)	Refine the System Definition (R)
		Change Request (C)	Manage Changing Requirements (R)
		Configuration Management Plan (C)	Manage Change Requests (C)
		Configuration Audit Findings (C)	Analyze the Problem (R)
		Project Repository (C)	Understand Stakeholder Needs (R)
SP 1.4 Maintain Bidirectional Traceability of Requirements	Software Requirements Specification (R)	Define the System (R)	
	Requirements Management Plan (R)	Manage the Scope of the System (R)	
		Refine the System Definition (R)	
		Manage Changing Requirements (R)	
		Manage the Scope of the System (R)	
SP 1.5 Ensure Alignment Between Project Work and Requirements	Review Record (C)	Manage Changing Requirements (R)	
	Configuration Management Plan (C)	Create Project Configuration Management (CM) Environments (C)	
	Stakeholder Requests (R)	Plan Project Configuration & Change Control (C)	
	Change Request (C)	Monitor & Report Configuration Status (C)	
	Work Order (C)	Analyze the Problem (R)	
		Understand Stakeholder Needs (R)	
		Define the System (R)	
		Manage the Scope of the System (R)	
		Refine the System Definition (R)	
		Manage Change Requests (C)	
		Manage Baselines & Releases (C)	
		Change and Deliver Configuration Items (C)	

*Atividades e artefatos com (R) são da disciplina de Requisitos e com (C) são da disciplina de Configuração e Gerenciamento de Mudanças do RUP.

Tabela 11 – Associação da Área de Processo Gerenciamento de Configuração

CMMI: Level 2 - Process Area: Configuration Management (CM)			
Specific Goals CMMI	Specific Practices CMMI	Artifacts RUP	Activities RUP
SG 1 Establish Baselines	SP 1.1 Identify Configuration Items	Configuration Audit Findings Configuration Management Plan Work Order	Manage Change Requests (C) Plan Project Configuration & Change Control (C) Create Project Configuration Management (CM) Environments (C) Change and Deliver Configuration Items (C) Manage Baselines & Releases (C) Monitor & Report Configuration Status (C)
	SP 1.2 Establish a Configuration Management System	Configuration Management Plan Project Repository Configuration Audit Findings Change Request Workspace	Manage Change Requests (C) Plan Project Configuration & Change Control (C) Create Project Configuration Management (CM) Environments (C) Change and Deliver Configuration Items (C) Manage Baselines & Releases (C) Monitor & Report Configuration Status (C) Understand Stakeholder Needs (R) Define the System (R) Manage the Scope of the System (R) Manage Changing Requirements (R)
	SP 1.3 Create or Release Baselines	Configuration Management Plan Configuration Audit Findings	Plan Project Configuration & Change Control (C) Create Project Configuration Management (CM) Environments (C) Monitor & Report Configuration Status (C)
SG 2 Track and Control Changes	SP 2.1 Track Change Requests	Change Request Work Order Configuration Audit Findings	Manage Change Requests (C) Change and Deliver Configuration Items (C) Manage Baselines & Releases (C) Monitor & Report Configuration Status (C) Understand Stakeholder Needs (R) Define the System (R) Manage the Scope of the System (R) Manage Changing Requirements (R)
	SP 2.2 Control Configuration Items	Project Repository Configuration Audit Findings Workspace Change Request	Manage Change Requests (C) Create Project Configuration Management (CM) Environments (C) Change and Deliver Configuration Items (C) Manage Baselines & Releases (C) Monitor & Report Configuration Status (C) Understand Stakeholder Needs (R) Define the System (R) Manage the Scope of the System (R) Manage Changing Requirements (R)
SG 3 Establish Integrity	SP 3.1 Establish Configuration Management Records	Test Log Test Results Project Repository	Manage Change Requests (C) Create Project Configuration Management (CM) Environments (C) Change and Deliver Configuration Items (C) Manage Baselines & Releases (C) Monitor & Report Configuration Status (C)
	SP 3.2 Perform Configuration Audits	Configuration Audit Findings	Monitor & Report Configuration Status (C)

APÊNDICE B – ATIVIDADES E ELEMENTOS DE PROCESSO

Figura 33 – Atividade *Understand Stakeholder Needs* e elementos de processo

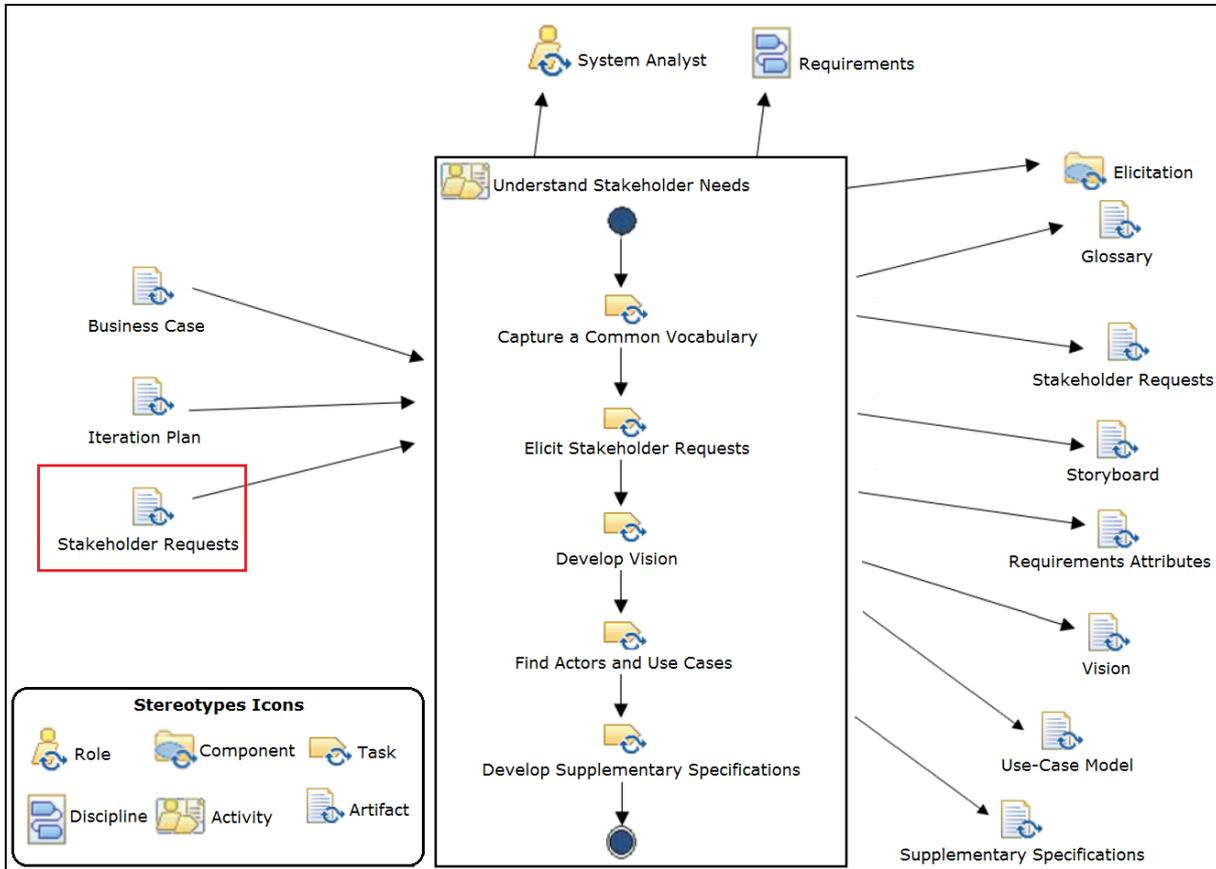


Figura 34 – Atividade *Refine the System Definition* e elementos de processo

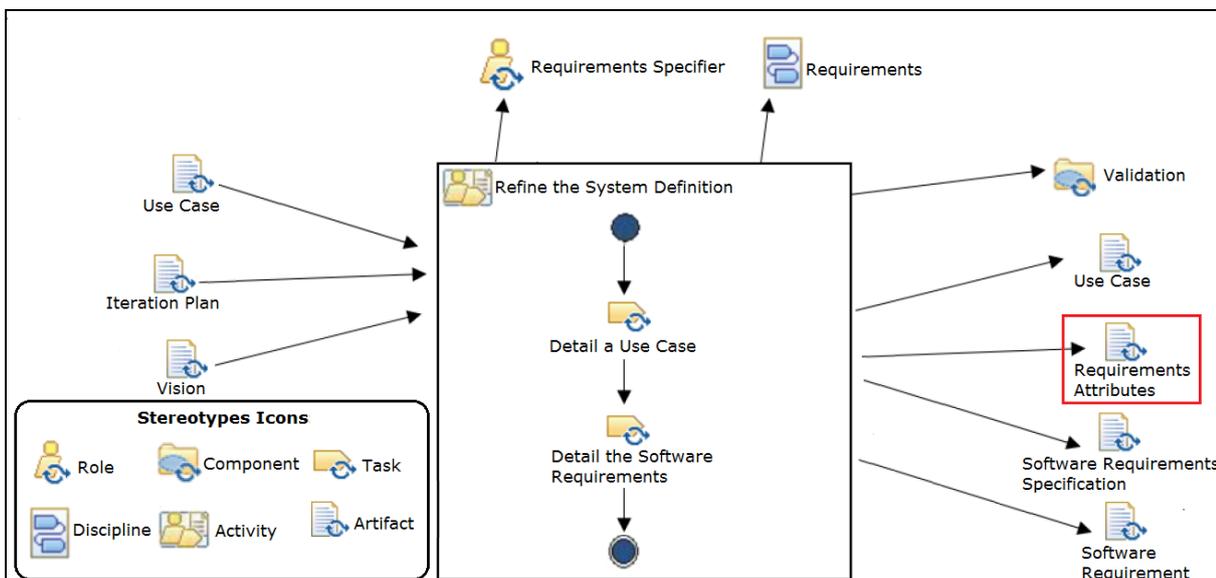


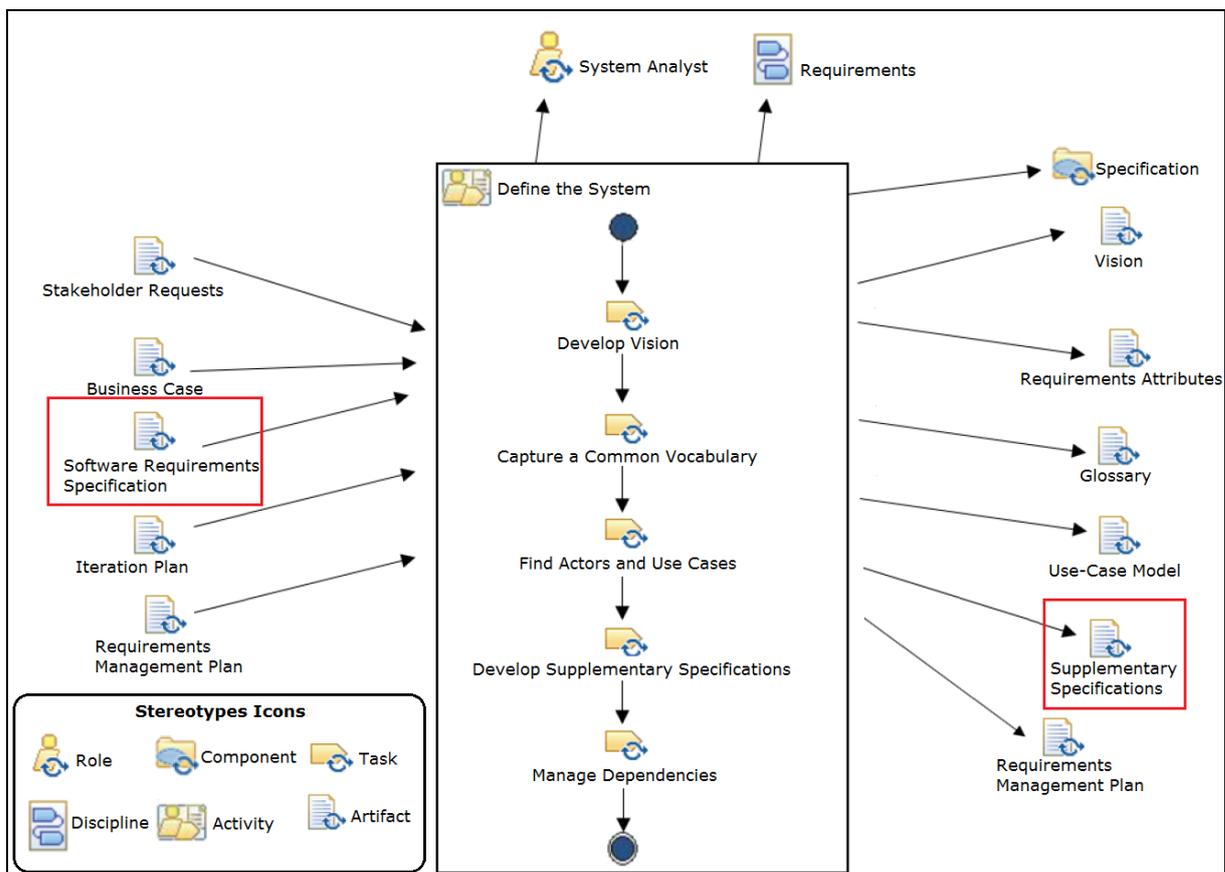
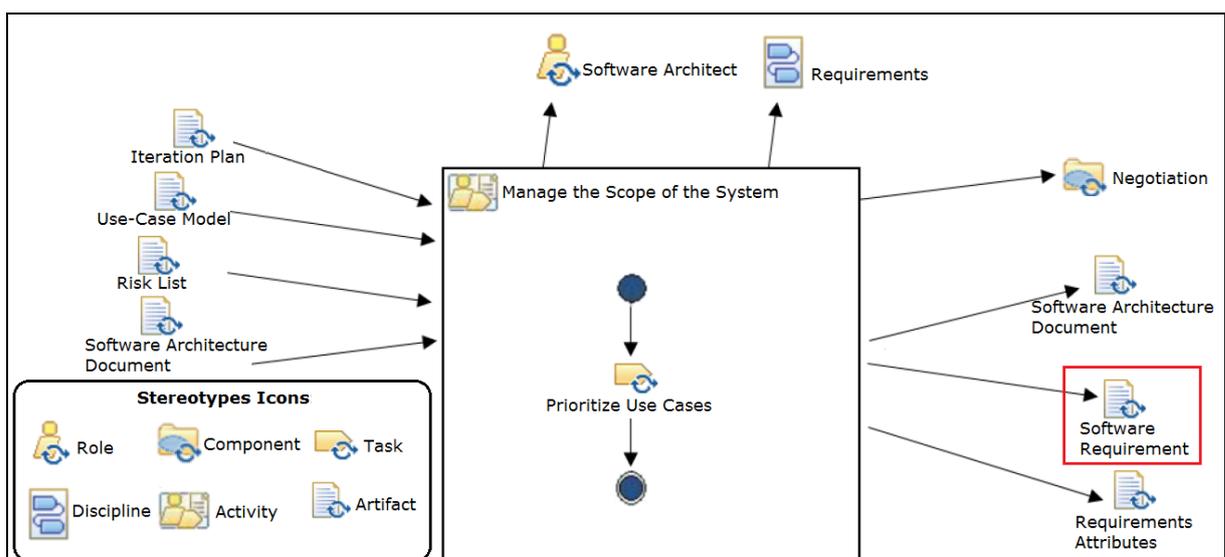
Figura 35 – Atividade *Define the System* e elementos de processoFigura 36 – Atividade *Manage the Scope of the System* e elementos de processo

Figura 37 – Atividade *Manage Changing Requirements* e elementos de processo

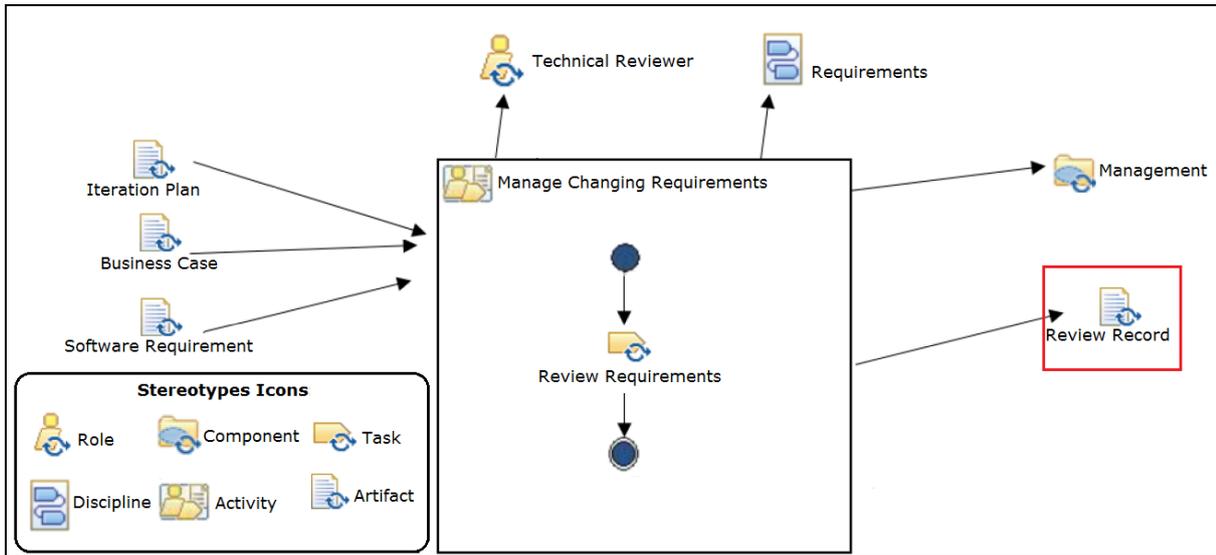


Figura 38 – Atividade *Plan Project Configuration & Change Control* e elementos de processo

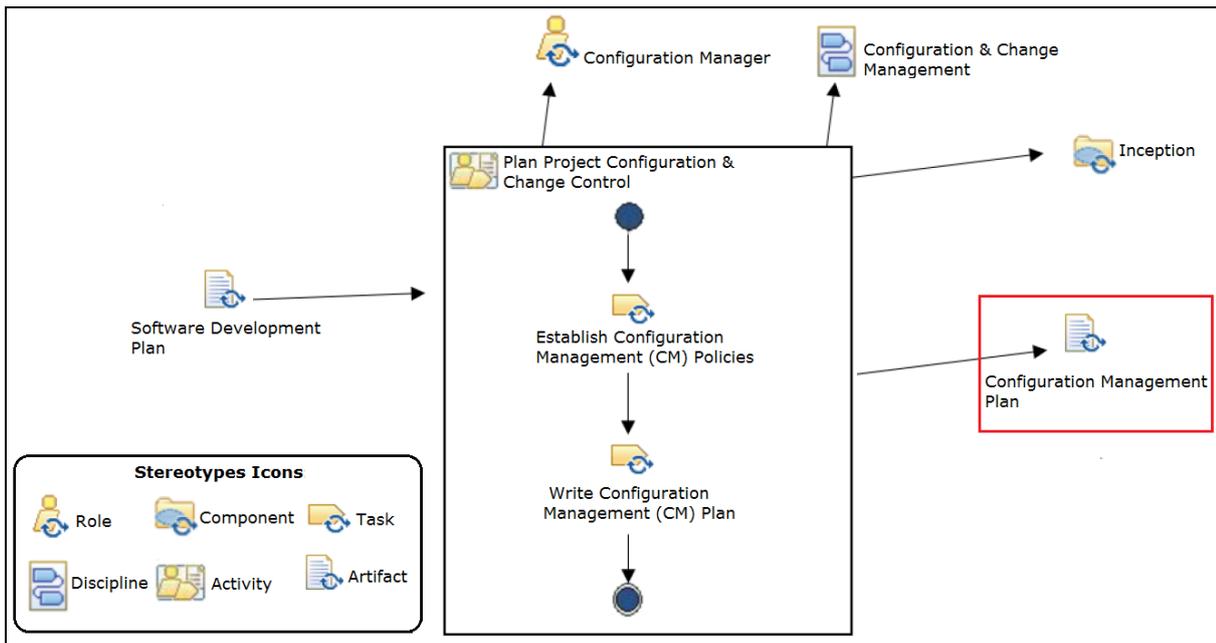


Figura 39 – Atividade *Manage Baselines & Releases* e elementos de processo

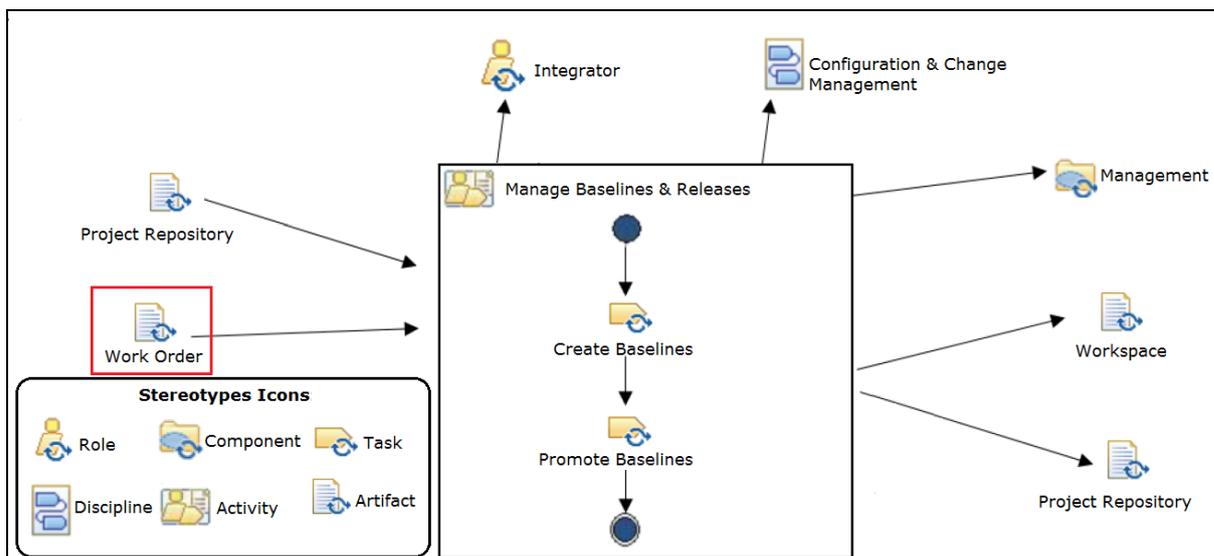


Figura 40 – Atividade *Create Project Configuration Management (CM) Environments* e elementos de processo

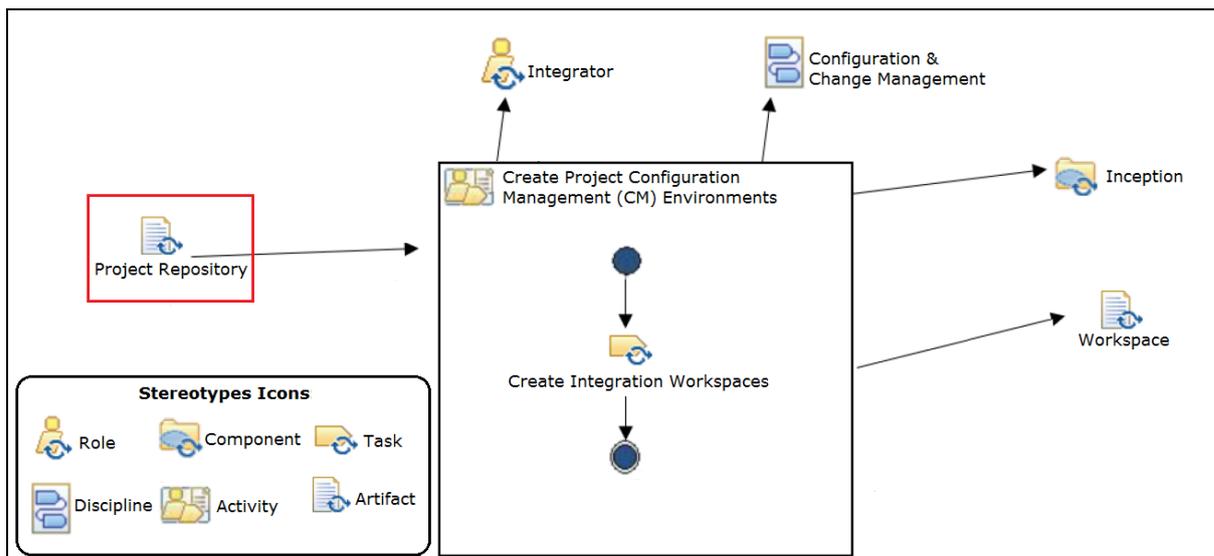


Figura 41 – Atividade *Monitor & Report Configuration Status* e elementos de processo

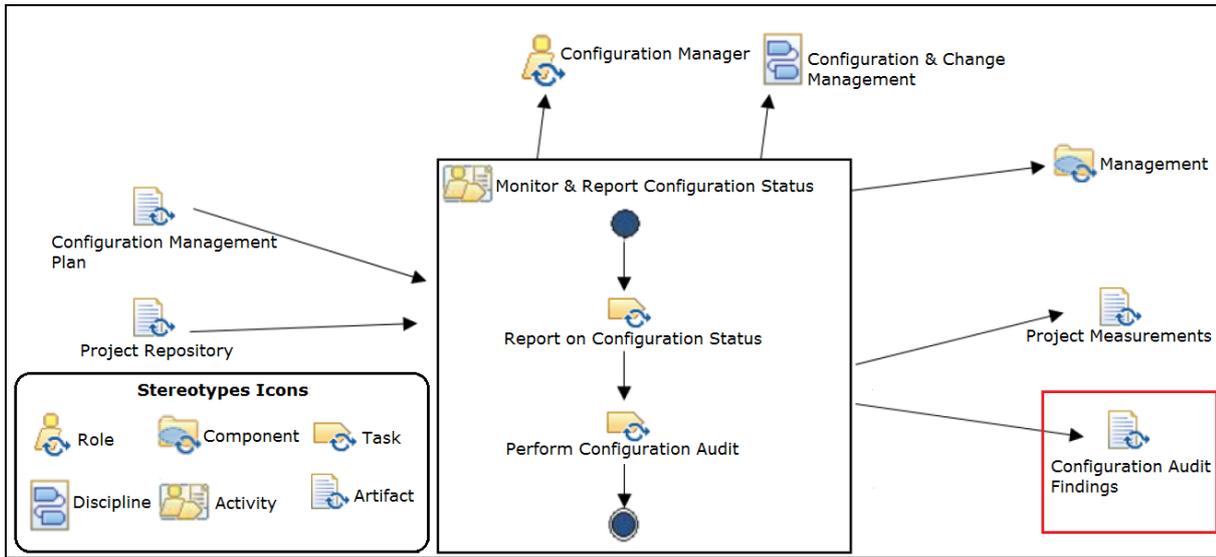


Figura 42 – Atividade *Change and Deliver Configuration Items* e elementos de processo

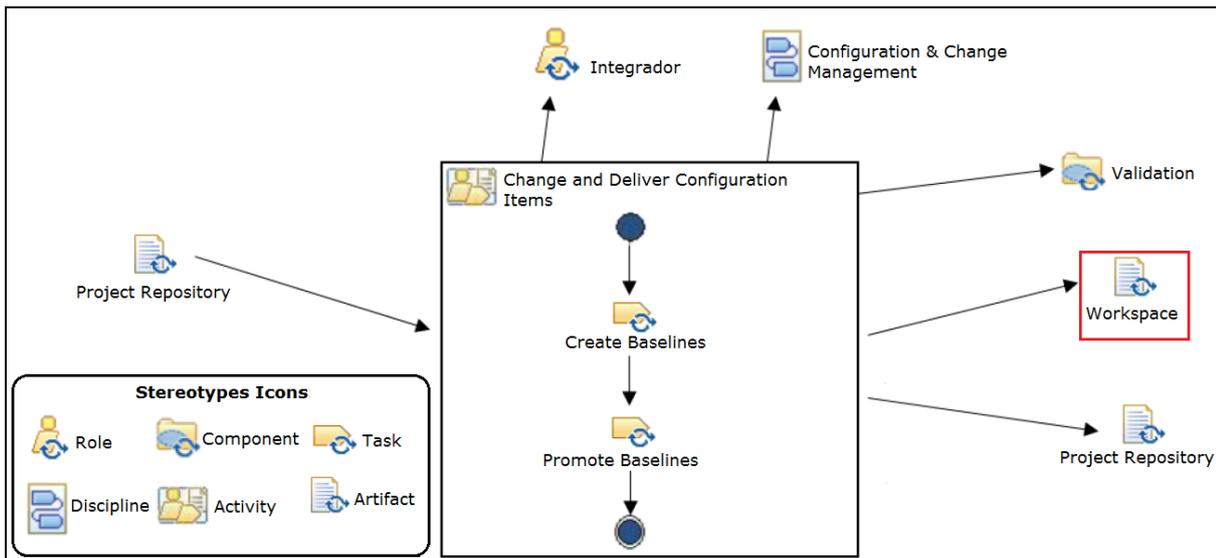
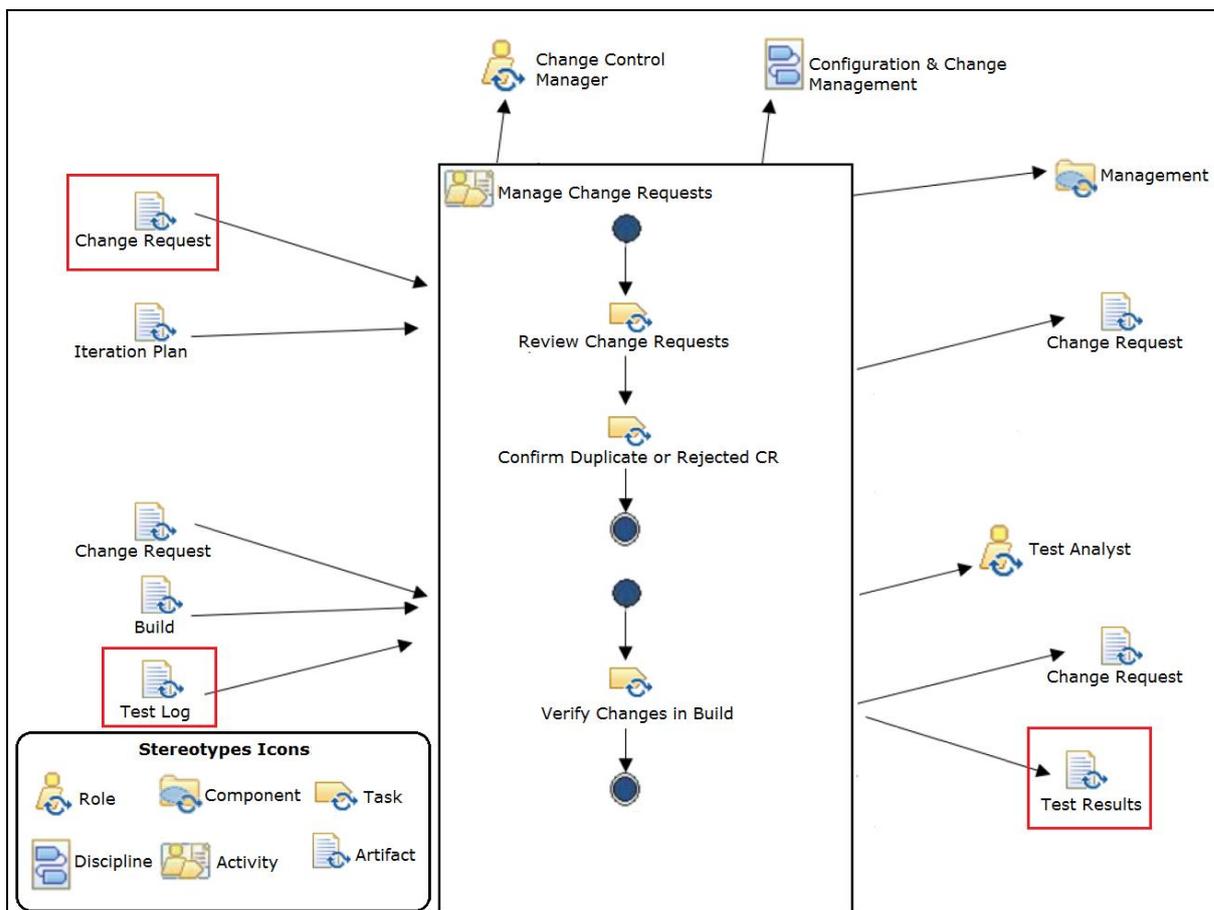


Figura 43 – Atividade *Manage Change Requests* e elementos de processo



APÊNDICE C – PLANOS DE QUALIDADE

Alguns artefatos possuem tanto para as práticas da área de processo de Gerenciamento de Requisitos, quanto para as práticas de Gerenciamento de Configuração do CMMI. Nestes casos, no plano de qualidade, os dados do RUP e a avaliação do artefato continuam iguais, o que é alterado são os dados do CMMI.

Tabela 12 – Plano de Qualidade: Artefato Plano de Gerenciamento de Configuração

PLANO DE QUALIDADE	
Dados do CMMI	
Área de Processo:	Gerenciamento de Requisitos Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos
Prática Específica:	SP 1.3 Gerenciar Mudanças nos Requisitos SP 1.5 Identificar Inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos
Dados do RUP	
Disciplina:	Configuração e Gerenciamento de Mudanças
Atividade:	Planejar Configuração do Projeto e Controle de Mudanças
Tarefa:	Escrever Plano de Gerenciamento de Configuração
Papel:	Gerente de Configuração
Avaliação do Artefato	
Artefato:	Plano de Gerenciamento de Configuração
Objetivo:	Detalhar o cronograma de atividades, as responsabilidades e os recursos necessários.
Meta de Qualidade:	Usabilidade Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Compreensibilidade
Método de Avaliação:	Completeness da documentação
Métrica:	$X = \text{número de atividades descritas} / \text{número total de atividades no sistema}$
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)
SubMeta de Qualidade:	Capacidade de aprendizagem
Método de Avaliação:	Frequência de ajuda
Métrica:	$X = \text{número de acessos para ajudar o desenvolvedor concluir a tarefa}$
Limite:	$0 \leq X$ (quanto mais próximo de 0 é melhor)

Tabela 13 – Dados do CMMI para as Práticas da Área de Processo Gerenciamento de Configuração.

Dados do CMMI	
Área de Processo:	Gerenciamento de Configuração Nível de Maturidade: 2
Meta Específica:	SG 1. Estabelecer Baselines
Prática Específica:	SP 1.1 Identificar Itens de Configuração SP 1.2 Estabelecer um Sistema de Gestão de Configuração SP 1.3 Criar ou Liberar Baselines

Tabela 14 – Plano de Qualidade: Artefato Repositório do Projeto

PLANO DE QUALIDADE		
Dados do CMMI		
Área de Processo:	Gerenciamento de Requisitos	Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos	
Prática Específica:	SP 1.3 Gerenciar Mudanças nos Requisitos	
Dados do RUP		
Disciplina:	Configuração e Gerenciamento de Mudanças	
Atividade:	Criar Projeto de Ambientes de Gerenciamento de Configuração	
Tarefa:	Criar Integração de Workspaces	
Papel:	Integrador	
Avaliação do Artefato		
Artefato:	Repositório do Projeto	
Objetivo:	Armazenar todas as versões de arquivos de projetos e diretórios.	
Meta de Qualidade:	Confiabilidade	Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Recuperabilidade	
Método de Avaliação:	Restaurabilidade	
Métrica:	X = número de casos de restauração feitos com sucesso/número de casos de restauração testados por requisitos	
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)	
Meta de Qualidade:	Manutenibilidade	
SubMeta de Qualidade:	Analisabilidade	
Método de Avaliação:	Eficiência na análise de falhas	
Métrica:	X = sum (t out - t in)/falhas registradas	
Limite:	0<=X (quanto mais próximo de 0 é melhor)	

Tabela 15 – Dados do CMMI para as Práticas da Área de Processo Gerenciamento de Configuração.

Dados do CMMI		
Área de Processo:	Gerenciamento de Configuração	Nível de Maturidade: 2
Meta Específica:	SG 1. Estabelecer Baselines	
	SG 2. Acompanhar e Controlar Mudanças	
	SG 3. Estabelecer Integridade	
Prática Específica:	SP 1.2 Estabelecer um Sistema de Gestão de Configuração	
	SP 2.2 Controlar Itens de Configuração	
	SP 3.1 Estabelecer Registros de Gestão de Configuração	

Tabela 16 – Plano de Qualidade: Artefato Registro de Auditoria de Configuração

PLANO DE QUALIDADE		
Dados do CMMI		
Área de Processo:	Gerenciamento de Requisitos	Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos	
Prática Específica:	SP 1.3 Gerenciar Mudanças nos Requisitos	
Dados do RUP		
Disciplina:	Configuração e Gerenciamento de Mudanças	
Atividade:	Monitorar e Relatar Status de Configuração	
Tarefa:	Realizar Auditoria de Configuração	
Papel:	Gerente de Configuração	
Avaliação do Artefato		
Artefato:	Registro de Auditoria de Configuração	
Objetivo:	Verificar se o desempenho do software está em conformidade com as suas exigências.	
Meta de Qualidade:	Funcionalidade	Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Funcionalidade	
Método de Avaliação:	Estabilidade de especificação funcional	
Métrica:	X = 1 - (número de funções alteradas depois de ter sido colocado em operação/número de funções especificadas)	
Limite:	0 ≤ X ≤ 1 (quanto mais próximo de 1,0 é melhor)	
SubMeta de Qualidade:	Conformidade	
Método de Avaliação:	Conformidade funcional	
Métrica:	X = 1 - (número de itens de conformidade funcional especificados que não foram implementadas durante o teste/número total de itens de conformidade funcional)	
Limite:	0 ≤ X ≤ 1 (quanto mais próximo de 1,0 é melhor)	

Tabela 17 – Dados do CMMI para as Práticas da Área de Processo Gerenciamento de Configuração.

Dados do CMMI		
Área de Processo:	Gerenciamento de Configuração	Nível de Maturidade: 2
Meta Específica:	SG 1. Estabelecer Baselines SG 2. Acompanhar e Controlar Mudanças SG 3. Estabelecer Integridade	
Prática Específica:	SP 1.1 Identificar Itens de Configuração SP 1.2 Estabelecer um Sistema de Gestão de Configuração SP 1.3 Criar ou Liberar Baselines SP 2.1 Acompanhar Solicitações de Mudança SP 2.2 Controlar Itens de Configuração SP 3.2 Executar Auditorias de Configuração	

Tabela 18 – Plano de Qualidade: Artefato Ordem de Serviço

PLANO DE QUALIDADE		
Dados do CMMI		
Área de Processo:	Gerenciamento de Requisitos	Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos	
Prática Específica:	SP 1.5 Identificar Inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos	
Dados do RUP		
Disciplina:	Configuração e Gerenciamento de Mudanças	
Atividade:	Gerenciar Baselines e Entregas	
Tarefa:	Criar Baselines	
Papel:	Integrador	
Avaliação do Artefato		
Artefato:	Ordem de Serviço	
Objetivo:	Transformar o planejamento em ação dentro de iterações.	
Meta de Qualidade:	Usabilidade	Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Conformidade	
Método de Avaliação:	Conformidade usabilidade	
Métrica:	X = 1 - (número de itens de conformidade de usabilidade especificadas que não foram implementadas durante o teste/número total de itens de conformidade usabilidade)	
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)	
SubMeta de Qualidade:	Operacionalidade	
Método de Avaliação:	Consistência operacional em uso	
Métrica:	X = 1 - (número de mensagens ou funções que o usuário encontra/número de mensagens ou funções)	
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)	

Tabela 19 – Dados do CMMI para as Práticas da Área de Processo Gerenciamento de Configuração.

Dados do CMMI		
Área de Processo:	Gerenciamento de Configuração	Nível de Maturidade: 2
Meta Específica:	SG 1. Estabelecer Baselines	
	SG 2. Acompanhar e Controlar Mudanças	
Prática Específica:	SP 1.1 Identificar Itens de Configuração	
	SP 2.1 Acompanhar Solicitações de Mudança	

Tabela 20 – Plano de Qualidade: Artefato Registro de Revisão

PLANO DE QUALIDADE		
Dados do CMMI		
Área de Processo:	Gerenciamento de Requisitos	Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos	
Prática Específica:	SP 1.5 Identificar Inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos	
Dados do RUP		
Disciplina:	Requisitos	
Atividade:	Gerenciar Requisitos Alterados	
Tarefa:	Revisar Requisitos	
Papel:	Revisor Técnico	
Avaliação do Artefato		
Artefato:	Registro de Revisão	
Objetivo:	Capturar os resultados de uma atividade de revisão em que os artefatos são revisados.	
Meta de Qualidade:	Manutenibilidade	Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Estabilidade	
Método de Avaliação:	Razão de modificações bem sucedidas	
Métrica:	X = número de casos que usuário encontra falhas durante a operação depois que o software foi alterado/tempo de operação durante o período de observação especificado após software ser alterado	
Limite:	0<=X (quanto mais próximo de 0 é melhor)	
SubMeta de Qualidade:	Analisabilidade	
Método de Avaliação:	Suporte ao diagnóstico	
Métrica:	X = falhas diagnosticadas com funções de diagnóstico/falhas registradas	
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)	

Tabela 21 – Plano de Qualidade: Artefato Atributos de Requisitos

PLANO DE QUALIDADE		
Dados do CMMI		
Área de Processo:	Gerenciamento de Requisitos	Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos	
Prática Específica:	SP 1.1 Obter Entendimento dos Requisitos	
Dados do RUP		
Disciplina:	Requisitos	
Atividade:	Aperfeiçoar a Definição do Sistema	
Tarefa:	Detalhar Requisitos do Software	
Papel:	Especificador de Requisitos	
Avaliação do Artefato		
Artefato:	Atributos de Requisitos	
Objetivo:	Descrever um repositório de requisitos do projeto, atributos e dependências.	
Meta de Qualidade:	Funcionalidade	Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Adequação	
Método de Avaliação:	Adequação funcional	
Métrica:	X = 1 - (número de funções nas quais são detectados problemas em avaliação/número de funções avaliadas)	
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)	
SubMeta de Qualidade:	Conformidade	
Método de Avaliação:	Conformidade funcional	
Métrica:	X = 1 - (número de itens de conformidade funcionalidade especificados que não foram implementadas durante o teste/número total de itens de conformidade funcionalidade especificados)	
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)	

Tabela 22 – Plano de Qualidade: Artefato Especificações Suplementares

PLANO DE QUALIDADE		
Dados do CMMI		
Área de Processo:	Gerenciamento de Requisitos	Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos	
Prática Específica:	SP 1.1 Obter Entendimento dos Requisitos	
Dados do RUP		
Disciplina:	Requisitos	
Atividade:	Definir o Sistema	
Tarefa:	Desenvolver Especificações Suplementares	
Papel:	Analista de Sistemas	
Avaliação do Artefato		
Artefato:	Especificações Suplementares	
Objetivo:	Capturar requisitos que não são capturados imediatamente no modelo de casos de uso.	
Meta de Qualidade:	Funcionalidade	Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Adequação	
Método de Avaliação:	Compleitude de implementação funcional	
Métrica:	$X = 1 - (\text{número de funções ausentes detectadas na avaliação} / \text{número de funções especificadas})$	
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)	
Meta de Qualidade:	Confiabilidade	
SubMeta de Qualidade:	Recuperabilidade	
Método de Avaliação:	Restaurabilidade	
Métrica:	$X = \text{número de casos de restauração feitos com sucesso} / \text{número de casos de restauração testados por requisitos}$	
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)	

Tabela 23 – Plano de Qualidade: Artefato Requisitos de Software

PLANO DE QUALIDADE		
Dados do CMMI		
Área de Processo:	Gerenciamento de Requisitos	Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos	
Prática Específica:	SP 1.1 Obter Entendimento dos Requisitos	
Dados do RUP		
Disciplina:	Requisitos	
Atividade:	Gerenciar o Escopo do Sistema	
Tarefa:	Priorizar Casos de Uso	
Papel:	Arquiteto de Software	
Avaliação do Artefato		
Artefato:	Requisitos de Software	
Objetivo:	Especificar uma condição ou capacidade com a qual um sistema deve estar em conformidade.	
Meta de Qualidade:	Funcionalidade	Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Adequação	
Método de Avaliação:	Adequação funcional	
Métrica:	$X = 1 - (\text{número de funções nas quais são detectados problemas em avaliação} / \text{número de funções avaliadas})$	
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)	
SubMeta de Qualidade:	Conformidade	
Método de Avaliação:	Conformidade funcional	
Métrica:	$X = 1 - (\text{número de itens de conformidade funcionalidade especificados que não foram implementados durante o teste} / \text{número total de itens de conformidade funcionalidade especificados})$	
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)	

Tabela 24 – Plano de Qualidade: Artefato Especificação de Requisitos de Software

PLANO DE QUALIDADE	
Dados do CMMI	
Área de Processo:	Gerenciamento de Requisitos Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos
Prática Específica:	SP 1.1 Obter Entendimento dos Requisitos SP 1.4 Manter Rastreabilidade Bidirecional dos Requisitos
Dados do RUP	
Disciplina:	Requisitos
Atividade:	Definir o Sistema
Tarefa:	Capturar um Vocabulário Comum
Papel:	Analista de Sistemas
Avaliação do Artefato	
Artefato:	Especificação de Requisitos de Software
Objetivo:	Organizar todos os requisitos que envolvem o projeto.
Meta de Qualidade:	Funcionalidade Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Conformidade
Método de Avaliação:	Conformidade funcional
Métrica:	$X = 1 - (\text{número de itens de conformidade funcionalidade especificados que não foram implementados durante o teste} / \text{número total de itens de conformidade funcionalidade especificados})$
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)
Meta de Qualidade:	Usabilidade
SubMeta de Qualidade:	Compreensibilidade
Método de Avaliação:	Completeness da documentação
Métrica:	$X = \text{número de requisitos descritos} / \text{número total de requisitos do sistema}$
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)

Tabela 25 – Plano de Qualidade: Artefato Solicitação dos Stakeholders

PLANO DE QUALIDADE	
Dados do CMMI	
Área de Processo:	Gerenciamento de Requisitos Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos
Prática Específica:	SP 1.2 Obter Comprometimento com os Requisitos SP 1.3 Gerenciar Mudanças nos Requisitos SP 1.5 Identificar Inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos
Dados do RUP	
Disciplina:	Requisitos
Atividade:	Entender as Necessidades dos Stakeholders
Tarefa:	Elaborar Visão
Papel:	Analista de Sistemas
Avaliação do Artefato	
Artefato:	Solicitação dos Stakeholders
Objetivo:	Capturar todas as solicitações feitas no projeto.
Meta de Qualidade:	Manutenibilidade Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Analisabilidade
Método de Avaliação:	Capacidade de análise das falhas
Métrica:	$X = 1 - (\text{falhas sem causas identificadas} / \text{falhas registradas})$
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)
SubMeta de Qualidade:	Modificabilidade
Método de Avaliação:	Eficiência do ciclo de modificação
Métrica:	$X = \text{sum}(t_{\text{out}} - t_{\text{in}}) / \text{versões}$
Limite:	$0 < X$ (quanto mais curto é melhor)

Tabela 26 – Plano de Qualidade: Artefato Plano de Gerenciamento de Requisitos

PLANO DE QUALIDADE	
Dados do CMMI	
Área de Processo:	Gerenciamento de Requisitos Nível de Maturidade: 2
Meta Específica:	SG 1. Gerenciar Requisitos
Prática Específica:	SP 1.2 Obter Comprometimento com os Requisitos SP 1.4 Manter Rastreabilidade Bidirecional dos Requisitos
Dados do RUP	
Disciplina:	Requisitos
Atividade:	Analisar o Problema
Tarefa:	Elaborar Plano de Gerenciamento de Requisitos
Papel:	Analista de Sistemas
Avaliação do Artefato	
Artefato:	Plano de Gerenciamento de Requisitos
Objetivo:	Descrever como o projeto irá configurar e gerenciar os requisitos.
Meta de Qualidade:	Funcionalidade Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Conformidade
Método de Avaliação:	Conformidade funcional
Métrica:	X = 1 - (número de itens de conformidade funcionalidade especificados que não foram implementadas durante o teste/número total de itens de conformidade funcionalidade especificados)
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)
Meta de Qualidade:	Usabilidade
SubMeta de Qualidade:	Compreensibilidade
Método de Avaliação:	Completo da documentação
Métrica:	X = número de requisitos descritos/número total de requisitos do sistema
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)
SubMeta de Qualidade:	Capacidade de aprendizagem
Método de Avaliação:	Eficácia da documentação
Métrica:	X = número de tarefas concluídas com êxito após acessar o plano/total de números de tarefas testadas
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)

Tabela 27 – Plano de Qualidade: Artefato Resultados do Teste

PLANO DE QUALIDADE	
Dados do CMMI	
Área de Processo:	Gerenciamento de Requisitos Nível de Maturidade: 2
Meta Específica:	SG 3. Estabelecer Integridade
Prática Específica:	SP 3.1 Estabelecer Registros de Gestão de Configuração
Dados do RUP	
Disciplina:	Gerenciamento de Configuração e Mudanças
Atividade:	Gerenciar Pedido de Mudanças
Tarefa:	Verificar Mudanças no Build
Papel:	Analista de Teste
Avaliação do Artefato	
Artefato:	Resultados do Teste
Objetivo:	Resume a análise de um ou mais Logs de Teste e Pedidos de Mudança.
Meta de Qualidade:	Usabilidade Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Operacionalidade
Método de Avaliação:	Consistência operacional em uso
Métrica:	X = 1 - (número de mensagens ou funções que o usuário encontra/número de mensagens ou funções)
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)
SubMeta de Qualidade:	Conformidade
Método de Avaliação:	Conformidade usabilidade
Métrica:	X = 1 - (número de itens de conformidade de usabilidade especificadas que não foram implementadas durante o teste/número total de itens de conformidade usabilidade)
Limite:	0<=X<=1 (quanto mais próximo de 1,0 é melhor)

Tabela 28 – Plano de Qualidade: Artefato *Workspace*

PLANO DE QUALIDADE	
Dados do CMMI	
Área de Processo:	Gerenciamento de Requisitos Nível de Maturidade: 2
Meta Específica:	SG 1. Estabelecer Baselines
	SG 2. Acompanhar e Controlar Mudanças
Prática Específica:	SP 1.2 Estabelecer um Sistema de Gestão de Configuração
	SP 2.2 Controlar Itens de Configuração
Dados do RUP	
Disciplina:	Gerenciamento de Configuração e Mudança
Atividade:	Alterar e Entregar Itens de Configuração
Tarefa:	Criar Baselines
Papel:	Integrador
Avaliação do Artefato	
Artefato:	Workspace
Objetivo:	Fornecer acesso seguro e exclusivo para artefatos do projeto sob controle de versão.
Meta de Qualidade:	Confiabilidade Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Tolerância a falhas
Método de Avaliação:	Evitar operações incorretas
Métrica:	X = número de ocorrências de falhas críticas e graves/número de operações incorretas
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)
Meta de Qualidade:	Funcionalidade
SubMeta de Qualidade:	Segurança
Método de Avaliação:	Controle de acesso
Métrica:	X = número detectado de tipos diferentes de operações ilegais/número de tipos de operações ilegais especificadas
Limite:	$0 \leq X \leq 1$ (quanto mais próximo de 1,0 é melhor)

Tabela 29 – Plano de Qualidade: Artefato *Test Log*

PLANO DE QUALIDADE	
Dados do CMMI	
Área de Processo:	Gerenciamento de Requisitos Nível de Maturidade: 2
Meta Específica:	SG 3. Estabelecer Integridade
Prática Específica:	SP 3.1 Estabelecer Registros de Gestão de Configuração
Dados do RUP	
Disciplina:	Gerenciamento de Configuração e Mudanças
Atividade:	Gerenciar Pedido de Mudanças
Tarefa:	Verificar Mudanças no Build
Papel:	Analista de Teste
Avaliação do Artefato	
Artefato:	Test Log
Objetivo:	Fornecer a verificação de um conjunto de testes que foi executado.
Meta de Qualidade:	Funcionalidade Tipo de Qualidade: ISO/IEC 9126
SubMeta de Qualidade:	Acurácia
Método de Avaliação:	Precisão de expectativa
Métrica:	X = número de casos encontrados pelos usuários com uma diferença contra os resultados esperados razoáveis além permitido/tempo de operação
Limite:	$0 \leq X$ (quanto mais próximo de 0 é melhor)
Meta de Qualidade:	Eficiência
SubMeta de Qualidade:	Comportamento em relação ao tempo
Método de Avaliação:	Tempo de espera
Métrica:	X = tempo total gasto na espera/tempo da tarefa
Limite:	$0 \leq X$ (quanto menor é melhor)
Meta de Qualidade:	Manutenibilidade
SubMeta de Qualidade:	Testabilidade
Método de Avaliação:	Eficiência do reteste
Métrica:	X = sum (tempo de teste para assegurar falha resolvida)/falhas resolvidas
Limite:	$0 < X$ (quanto menor é melhor)

APÊNDICE D – EXEMPLOS DE MÉTODOS DE AVALIAÇÃO E MÉTRICAS

Tabela 30 – Métodos de Avaliação e Métricas de acordo com a ISO/IEC 9126

Meta de Qualidade	Submeta de Qualidade	Método de Avaliação	Métrica
Funcionalidade	Adequação	Adequação funcional	X=1-(número de funções nas quais são detectados problemas em avaliação/número de funções avaliadas)
		Completude de implementação funcional	X=1-(número de funções ausentes detectadas na avaliação/número de funções especificadas)
		Abrangência de implementação funcional	X=1-(número de funções implementadas incorretas ou ausentes/número de funções especificadas)
	Acurácia	Estabilidade de especificação funcional	X=1-(número de funções alteradas depois de ter sido colocado em operação/número de funções especificadas)
		Precisão de expectativa	X=número de casos encontrados pelos usuários com uma diferença contra os resultados esperados razoáveis além permitido/tempo de operação
		Acurácia computacional	X=número de cálculos imprecisos encontrado pelos usuários/tempo de operação
	Interoperabilidade	Precisão	X=número de resultados encontrados por usuários com nível de precisão diferente do necessário/tempo de operação
		Permutabilidade de dados (com base em formato de dados)	X=número de formatos de dados que são aprovados a serem trocadas com sucesso com outro software ou sistema durante os testes de trocas de dados/número total de formatos de dados que devem ser trocados
		Permutabilidade de dados (com base na tentativa de sucesso do usuário)	X=1-(número de casos em que o usuário não conseguiu trocar dados com outros softwares ou sistemas/número de casos em que usuário tentou trocar dados)
	Segurança	Auditabilidade de acesso	X=número de acessos do usuário gravados no histórico do banco de dados/número de acessos feitos durante a avaliação
		Controle de acesso	X=número detectados de tipos diferentes de operações ilegais/número de tipos de operações ilegais especificadas
		Prevenção da corrupção de dados	X=número de vezes que ocorreu corrupção de dados/período de tempo de operação
	Conformidade	Conformidade funcional	X=1-(número de itens de conformidade funcionalidade especificados que não foram implementadas durante o teste/número total de itens de conformidade funcionalidade especificados)
Conformidade padrão de interface		X=número de interfaces implementadas corretamente conforme especificado/número total de interfaces que exigem conformidade	
Confiabilidade	Maturidade	Resolução de falhas	X=número de falhas resolvidas/número total de falhas detectadas
		Densidade de falhas	X=número de falhas detectadas/tamanho do produto
		Remoção de falhas	X=número de falhas corrigidas/número total de falhas detectadas
	Tolerância a Falhas	Evitar operações incorretas	X=número de ocorrências de falhas críticas e graves/número de operações incorretas
		Tempo ocioso	X=número de breakdowns observadas/tempo total ocioso
	Recuperabilidade	Média de tempo de recuperação	X=Sum(tempo de recuperação)/número de casos de sistemas observado que entraram em recuperação
		Reinicialização	X=número de reinicializações durante o teste/número total de reinicializações
		Restaurabilidade	X=número de casos de restauração feitos com sucesso/número de casos de restauração testados por requisitos
	Conformidade	Restaurar eficácia	X=número de casos restaurado com êxito/número de casos realizados
		Conformidade confiabilidade	X=1-(número de itens de conformidade de confiabilidade especificadas que não foram implementadas durante o teste/número total de itens de conformidade confiabilidade especificados)

Usabilidade	<p>Compreensibilidade</p> <p>Capacidade de aprendizagem</p> <p>Operacionalidade</p> <p>Atratividade</p> <p>Conformidade</p>	<p>Completude da documentação</p> <p>Demonstração de acessibilidade</p> <p>Demonstração de acessibilidade de uso</p> <p>Eficácia da demonstração</p> <p>Funções evidentes</p> <p>Facilidade de aprendizagem da função</p> <p>Eficácia da documentação do usuário e/ou sistema de ajuda</p> <p>Frequência de ajuda</p> <p>Consistência operacional em uso</p> <p>Erro de conexão</p> <p>Mensagens de erro auto-explicativas</p> <p>Interação atrativa</p> <p>Conformidade usabilidade</p>	<p>X=número de atividades descritas/número total de atividades no sistema</p> <p>X=número de tutoriais que o usuário acessa com êxito/número de tutoriais disponíveis</p> <p>X=número de casos em que usuário obtém sucesso com a demonstração/número de casos em que usuário tenta ver a demonstração</p> <p>X=número de funções operadas com sucesso/número de tutoriais acessadas</p> <p>X=número de funções identificados pelo usuário/número total de funções</p> <p>X=tempo médio necessário para aprender a usar uma função corretamente</p> <p>X=número de tarefas concluídas com êxito após acessar ajuda e/ou documentação online/total de número de tarefas testadas</p> <p>X=número de acessos para ajudar o desenvolvedor concluir sua tarefa</p> <p>X=1-(número de mensagens ou funções que o usuário encontra/número de mensagens ou funções)</p> <p>X=tempo de conclusão para correção de erros da tarefa executada/tempo de correção de erros da tarefa executada</p> <p>X=número de condições de erro para o qual o usuário propõe a ação de recuperação correto/número de condições de erro testado</p> <p>Questionário para avaliar a atratividade da interface para usuários, após uma experiência de uso</p> <p>X=1-(número de itens de conformidade de usabilidade especificadas que não foram implementadas durante o teste/número total de itens de conformidade usabilidade especificados)</p>
Eficiência	<p>Comportamento em relação ao tempo</p> <p>Utilização de recursos</p> <p>Conformidade</p>	<p>Tempo de resposta</p> <p>Tempo de espera</p> <p>Utilização da memória</p> <p>Utilização de entrada/saída</p> <p>Conformidade eficiência</p>	<p>X=tempo de adquirir o resultado-tempo de entrada de comando</p> <p>X=tempo total gasto na espera/tempo de tarefa</p> <p>X=quantidade de memória utilizada/memória total</p> <p>X=número de buffers(calculados ou simulados)</p> <p>X=1-(número de itens de conformidade de eficiência especificadas que não foram implementadas durante o teste/número total de itens de conformidade eficiência especificados)</p>
Manutenibilidade	<p>Analisabilidade</p> <p>Modificabilidade</p>	<p>Rastreabilidade das falhas</p> <p>Suporte ao diagnóstico</p> <p>Capacidade de análise das falhas</p> <p>Eficiência na análise das falhas</p> <p>Capacidade da monitoração do status</p> <p>Eficiência do ciclo de modificação</p> <p>Tempo médio de modificação</p> <p>Complexidade da modificação</p> <p>Modificabilidade parametrizada</p> <p>Capac. controle de modificação</p> <p>Facilidade nas mudanças</p> <p>Impacto da mudança</p>	<p>X=falhas identificadas/falhas registradas</p> <p>X=falhas diagnosticadas com funções de diagnóstico/falhas registradas</p> <p>X=1-(falhas sem causas identificadas/falhas registradas)</p> <p>X=Sum (T out - T in)/falhas registradas</p> <p>X=1-(casos sem informações monitoração/casos tentados)</p> <p>X=Sum (Tout - Tin)/versões</p> <p>X=Sum (Tout - Tin)/falhas registradas e removidas</p> <p>X=Sum (Tempo modificação/Tam modificação)/modificações realizadas</p> <p>X=casos resolvidos via mudança de parâmetros/tentativas via mudança de parâmetros</p> <p>X=registros efetivados no log de modificação/registros planejados para modificações</p> <p>X=soma do tempo dedicado à mudança/tamanho do software modificado</p> <p>X=1-número de impactos detectados após as modificações/número de modificações feitas</p>

	Estabilidade	Razão de modificações bem sucedidas	$X = (\text{falhas após modificação} / \text{tempo após modificação}) / (\text{falhas prévias a modificação} / \text{tempo prévio})$
	Testabilidade	Localização do impacto Eficiência do reteste	$X = \text{falhas novas após modificação} / \text{falhas resolvidas}$ $X = \text{Sum (Tempo de teste para assegurar falha resolvida)} / \text{falhas resolvidas}$
	Conformidade	Restartabilidade do Teste Conformidade manutenibilidade	$X = \text{casos em que pausa e recomeça o teste em pontos de checagem} / \text{casos identificados de pausas nos testes}$ $X = 1 - (\text{número de itens de conformidade de manutenibilidade especificadas que não foram implementadas durante o teste} / \text{número total de itens de conformidade manutenibilidade especificados})$
Portabilidade	Adaptabilidade	Adaptabilidade do ambiente de hardware	$X = 1 - (\text{número de funções operacionais de tarefas que não foram concluídas para atender níveis adequados durante o teste operacional combinado com o hardware} / \text{número total de funções que foram testados})$
	Capacidade de instalação	Esforço para instalação	$X = \text{número de manuais de usuário necessários para instalação}$
	Coexistência	Coexistência disponível	$X = \text{número de quaisquer restrições ou falhas inesperadas que usuário encontrou durante o funcionamento em simultâneo com outros softwares} / \text{tempo de duração de operações concomitantemente}$
	Capacidade de substituição	Inclusão da função	$X = \text{número de funções que produzem resultados semelhantes} / \text{número de funções testadas que são semelhantes às funções fornecidas por outro software a ser substituído}$
	Conformidade	Conformidade portabilidade	$X = 1 - (\text{número de itens de conformidade de portabilidade especificadas que não foram implementadas durante o teste} / \text{número total de itens de conformidade portabilidade especificados})$

APÊNDICE E – ETAPAS DO ESTUDO DE CASO

ETAPA 1 – Coleta de Dados do Participante e do Projeto

1. Nome: _____
2. Último curso que você concluiu:
 Graduação Especialização Mestrado Doutorado
3. Nome da empresa que trabalha: _____
4. Porte da empresa:
 Microempresa (até 9 empregados)
 Pequeno Porte (de 10 a 49 empregados)
 Médio Porte (de 50 a 99 empregados)
 Grande Porte (mais de 99 empregados)
5. Tempo em que está na empresa:
 menos de 1 ano de 1 a 3 anos de 4 a 6 anos de 7 a 10 anos mais de 10 anos
6. Grau de experiência:

Área de Conhecimento	Grau de Experiência				
	0	1	2	3	4
Engenharia de Software					
Gerência de Projetos de Software					
Qualidade de Software					
Adaptação de Processos					
Ferramentas de Adaptação de Processos					
Artefatos de Software					
Metodologia Ágil					
Metodologia Planejada					
Métricas de Software					
Modelo de Qualidade CMMI					

0 = nenhum

1 = apenas possui conhecimento teórico

2 = possui conhecimento teórico aplicado apenas no contexto acadêmico

3 = possui conhecimento teórico somado de experiências práticas individuais

4 = possui conhecimento teórico somado de experiências práticas reais

7. Nome do projeto: _____
8. Cargo que assume neste projeto: _____
9. Contexto do projeto:
- a. Tamanho da equipe:
 Pequena(até 12 pessoas) Média(13 a 50 pessoas) Grande(mais de 50 pessoas)

b. Arquitetura estável:

Estável Modificada Nova

c. Modelo de negócio:

Sob medida Comercial Componente de um grande sistema

d. Distribuição da equipe:

Local Equipes diferentes Distribuição geográfica

e. Taxa de mudanças:

Mais que 30% no mês Entre 10% e 30% no mês Menos que 10% no mês

f. Idade do sistema:

Novo desenvolvimento Manutenção Evolução de sistema legado

g. Criticidade:

Perda de conforto Perda de dinheiro Mortes

h. Controle:

Dinâmico/Flexível Regras Simples Mecânico/Formal

10. Requisito de Adaptação: Práticas do CMMI

a. Área de Processo: Gerenciamento de Requisitos (Nível 2)

SG 1 Gerenciar Requisitos

SP 1.1 Obter Entendimento dos Requisitos

SP 1.2 Obter Comprometimento com os Requisitos

SP 1.3 Gerenciar Mudanças nos Requisitos

SP 1.4 Manter Rastreabilidade Bidirecional dos Requisitos

SP 1.5 Identificar Inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos

b. Área de Processo: Gerenciamento de Configuração (Nível 2)

SG 1 Estabelecer *Baselines*

SP 1.1 Identificar Itens de Configuração

SP 1.2 Estabelecer um Sistema de Gestão de Configuração

SP 1.3 Criar ou Liberar *Baselines*

SG 2 Acompanhar e Controlar Mudanças

SP 2.1 Acompanhar Solicitações de Mudança

SP 2.2 Controlar Itens de Configuração

SG 3 Estabelecer Integridade

SP 3.1 Estabelecer Registros de Gestão de Configuração

SP 3.2 Executar Auditorias de Configuração

ETAPA 3 – Entrevista com Analistas de Qualidade
--

I. Processos de Software

1. Como foi elaborado o processo de software usado atualmente?
2. Em todos os projetos é utilizado o mesmo processo?
 - a. Caso a resposta seja sim, o processo atende as diferentes características de todos os projetos?
 - b. Caso a resposta seja não, como é efetuada a adaptação?
3. Você utiliza ou já utilizou processo ágil em algum dos projetos? Se sim, qual?
4. Você utiliza ou já utilizou processo planejado em algum dos projetos? Se sim, qual?
5. É importante considerar o contexto do projeto para a definição do processo?
6. Quais artefatos este projeto possui?
7. O processo gerado utilizando a Linha de Processo de Software foi satisfatório?

II. Qualidade de Software

1. Você utiliza ou já utilizou algum modelo de qualidade em algum dos projetos? Se sim, qual?
2. Você acha que se um artefato for defeituoso, existe o risco de atrasar o projeto ou impactar diretamente nos custos do projeto?
3. Você acha que selecionando práticas do CMMI, irá aumentar a qualidade do processo gerado?

III. Planos de Qualidade de Artefatos

1. No seu ponto de vista, é possível avaliar um artefato utilizando as informações apresentadas no plano de qualidade?
 - a. Caso a resposta seja sim, você acredita que possa minimizar a elaboração de artefatos defeituosos?
2. Em que tipo de projeto ou equipe você considera que os planos de qualidade sejam mais eficientes?
3. O formato de documentação proposto foi de fácil entendimento, auxiliando na descrição dos elementos de qualidade? Qual o grau de dificuldade na interpretação dos planos de qualidade?

4. A possibilidade de alterar os planos de qualidade gerados ajuda o analista de qualidade a selecionar outros elementos de qualidade que acredita ser mais relacionado?
5. Você possui alguma sugestão para melhoria da abordagem, algum comentário adicional (dificuldades, críticas e/ou sugestões) a respeito do estudo executado?