



Pós-Graduação em Ciência da Computação

**DESENVOLVIMENTO BASEADO EM MODELOS DE
FERRAMENTAS PARA AVALIAÇÃO DA ADERÊNCIA DE
PROCESSOS DE SOFTWARE EM RELAÇÃO A
MODELOS DE MATURIDADE**

Por

Luiz Felipe de Oliveira Libório

Dissertação de Mestrado



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE

2014



Universidade Federal de Pernambuco

Centro de Informática

Pós-graduação em Ciência da Computação

Luiz Felipe de Oliveira Libório

**DESENVOLVIMENTO BASEADO EM MODELOS DE
FERRAMENTAS PARA AVALIAÇÃO DA ADERÊNCIA DE
PROCESSOS DE SOFTWARE EM RELAÇÃO A
MODELOS DE MATURIDADE**

*Trabalho apresentado ao Programa de Pós-graduação em
Ciência da Computação do Centro de Informática da Univer-
sidade Federal de Pernambuco como requisito parcial para
obtenção do grau de Mestre em Ciência da Computação.*

Orientador: *Alexandre Marcos Lins de Vasconcelos*

RECIFE

2014

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

L696d Libório, Luiz Felipe de Oliveira
Desenvolvimento baseado em modelos de ferramentas para avaliação da
aderência de processos de software em relação a modelos de maturidade /
Luiz Felipe de Oliveira Libório. – 2014.
106 f.: il., fig., tab.

Orientador: Alexandre Marcos Lins de Vasconcelos.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn,
Ciência da Computação, Recife, 2014.

Inclui referências e apêndices.

1. Engenharia de software. 2. Melhoria de processo de software. 3.
Qualidade de software. I. Vasconcelos, Alexandre Marcos Lins de (orientador).
II. Título.

005.1

CDD (23. ed.)

UFPE- MEI 2016-066

Dissertação de Mestrado apresentada por **Luiz Felipe de Oliveira Libório** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**Desenvolvimento Baseado em Modelos de Ferramentas para Avaliação da Aderência de Processos de Software em Relação a Modelos de Maturidade**” orientada pelo **Prof. Alexandre Marcos Lins de Vasconcelos** e aprovada pela Banca Examinadora formada pelos professores:

Prof. Hermano Perrelli de Moura
Centro de Informática/UFPE

Profa. Cristine Martins Gomes de Gusmão
Departamento de Engenharia Biomédica / UFPE

Prof. Alexandre Marcos Lins de Vasconcelos
Centro de Informática /UFPE

Visto e permitida a impressão.
Recife, 27 de agosto de 2014.

Profa. Edna Natividade da Silva Barros
Coordenadora da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

*Dedico esta dissertação a toda a minha família, amigos,
professores, colegas de trabalho e todos que tiveram
paciência e me deram o suporte necessário para chegar até
aqui.*

Agradecimentos

Em primeiro lugar, gostaria de agradecer à Deus, que sempre esteve presente em minha vida e durante este mestrado. Nos bons e nos maus momentos, sempre pude contar com Sua força para me guiar e não desistir, mesmo quando parecia a única alternativa.

Também gostaria de agradecer muito à minha família. Meu pai, Alcides, minha mãe, Zeneide, à minha irmã Carol, à minha sobrinha Ana Clara e à Ana Cláudia. Mesmo quase sempre sem saber o que eu fazia, estavam me apoiando, cada um a seu modo e são diretamente responsáveis por tudo isto. Devo tudo que tenho a eles, e nada mais justo do que agradecê-los neste momento tão importante para mim.

Gostaria também de agradecer a meu orientador, Prof. Alexandre. À sua enorme paciência, aos valiosos conselhos e por toda a orientação fornecida, além de toda solicitude em me atender e questionar sobre o trabalho, me fizeram chegar até aqui a isso sou muito grato. O que eu aprendi durante este tempo com certeza vou levar para o resto da vida, pois reconheço nele além de um grande professor, um grande ser humano, dos que só se conhece poucas vezes na vida.

Agradeço aos amigos do Teia-nela, turma que está junta desde o início da graduação, em 2007 e vai continuar junta sempre. Muitos, como eu, passaram pela difícil trajetória do mestrado e, mesmo sem saber, me motivaram com sua dedicação e persistência a prosseguir e atingir meus objetivos. Desejo a todos o sucesso que merecem e que ele seja na mesma medida da gratidão que tenho a todos vocês.

Agradeço também à minha namorada, Alice. Por compreender, aguentar, escutar, estar sempre comigo, me motivar ou simplesmente estar do meu lado em momentos difíceis, era tudo que eu poderia esperar dela e tive tudo isso de sobra.

Aos meus colegas de trabalho, que foram muitos durante este tempo, gostaria também de agradecer todo apoio e aprendizado que tive com eles durante este tempo. Pessoas como Luís Cláudio, Rogério, Arlindo, Paulo Gomes, Luiz Eugênio (Left), Paulo de Tarso e Eduardo Rocha passam pela vida profissional e deixam a sua marca. Muito obrigado pelo privilégio de poder ter tido a experiência fantástica de conviver e aprender com todos.

Enfim, agradeço a todos os amigos e familiares que não foram citados, mas que tiveram sua contribuição, seja perguntando, seja motivando ou simplesmente estando por perto. Sem vocês isso teria sido muito mais difícil, para não dizer impossível.

Existem dois tipos de problemas: grandes e pequenos. Os problemas pequenos não existem. E os grandes, você transforma em pequenos.

—ANTIGO PROVÉRPIO CHINÊS

Resumo

Atualmente, é cada vez mais exigido como critério competitivo a aderência do processo de software de uma organização a um modelo ou norma de qualidade. Alcançar altos níveis de aderência é uma tarefa complexa para as organizações, pois a atividade de avaliar essa aderência exige cuidado ao ser realizada. Com isso, busca-se cada vez mais o apoio de ferramentas de avaliação computadorizadas, cujo objetivo é agilizar e tornar mais confiável o processo de avaliação dos processos de software. Ainda assim, um dos fatores que tendem a dificultar a utilização e propagação do uso de ferramentas é a atualização das normas de qualidade e dos métodos de avaliação associados, que visam se adequar constantemente às boas práticas do mercado. Ora, se uma ferramenta é criada baseada na norma de qualidade e no seu método de avaliação, qualquer mudança em algum dos dois componentes significa mudanças na ferramenta, gerando custos de manutenção e de distribuição da nova versão da ferramenta. Este trabalho busca criar uma engine de geração que seja sensível a essas mudanças, sem que seu processo de desenvolvimento necessite ser executado novamente. Para isto, utiliza o processo MDD (Model-Driven Development) para atender a esses requisitos, dado que o seu objetivo é transformar elementos mais abstratos em elementos mais concretos utilizando uma sucessão de transformações, através de modelos. O objetivo deste trabalho é, então, aplicar as regras de MDD aos modelos de qualidade e métodos de avaliação. Isto torna possível a criação de ferramentas para avaliação baseada nos modelos necessários à avaliação. Ou seja, através de um modelo de processo e de um metamodelo da norma de qualidade, além do método de avaliação, é possível criar de forma automática via MDD uma ferramenta que possa ser atualizada sob demanda, sem que seu código-fonte precise ser alterado manualmente.

Palavras-chave: Normas de Qualidade. Métodos de Avaliação. Avaliação de Processos de Software. MDD. Model-Driven Development

Abstract

Nowadays, it is increasingly required as a competitive criteria the adherence of an organization to a software model or quality standard process. Achieving high levels of adherence is a complex task for organizations, because the activity of evaluating this adherence requires care to be performed. Thus, it is increasing the need of computerized evaluation tools whose goal is to speed up and make more reliable the process of evaluating software processes. Still, one of the factors that tend to hamper the use and propagation of tool use is the change of the quality standards and assessment methods, which aim to adapt constantly to best practices in the market. So, if a tool is created based on quality standard and its evaluation method, any change in any of the two components mean changes in the tool, generating costs of maintaining and distributing the new version of the tool. This work seeks to develop a tool that is sensitive to that changes, but without the need of performing its development process again. For that, it uses MDD (Model-Driven Development) processes meet these requirements, since their goal is to transform more abstract elements into more concrete elements using a succession of transformations, through models. So, the goal of this work is to apply the rules of MDD to quality models and assessment methods. It makes possible to create tools based on models. In other words, through a process model and a metamodel quality standard, besides the evaluation method, it is possible to create automatically via MDD a tool that can be updated on demand, without its source code need to be changed manually.

Keywords: Quality Standards. Assessment Methods. Software Process Evaluation. MDD. Model-Driven Development

Lista de Figuras

2.1	Evolução histórica do CMMI	23
2.2	Componentes do modelo do CMMI.	24
2.3	Estrutura das Representações	26
2.4	Organização do Modelo MPS com seus componentes e artefatos	27
2.5	Comparação entre as classes ARC	31
2.6	Relacionamento entre os modelos do MDD	38
3.1	Arquitetura do ambiente ImPProS	43
3.2	Ferramenta ProEvaluator - Tela Geral de Avaliação	45
3.3	Ferramenta ProEvaluator - Tela de Avaliação de Processo	45
3.4	Ferramenta ProEvaluator - Tela de Gráficos dos Resultados Esperados do Processo	46
3.5	Ferramenta ProEvaluator 2.0- Fluxo de Avaliação	47
3.6	Ferramenta ProEvaluator - Tela de Cadastro de Framework	48
3.7	Ferramenta ProEvaluator - Tela de Cadastro de Modelo	48
3.8	Ferramenta ProEvaluator - Tela de Avaliação	49
3.9	FAPS - Tela de Evidências	50
3.10	FAPS - Tela de Avaliação de Componentes do Processo	51
4.1	Atividades realizadas pela Engine	54
4.2	Workflow da Engine de Geração de Ferramentas	59
4.3	Workflow da Ferramenta Gerada	59
4.4	Módulos e Artefatos que compõem todo o ambiente da Engine	61
4.5	Relacionamento entre os modelos da Engine de acordo com o MOF	61
4.6	Modelo utilizado para os exemplos	66
4.7	Tela inicial da Engine	67
4.8	Menu da Engine	67
4.9	Dados da ferramenta ser gerada	68
4.10	Dados da norma de qualidade e do processo a ser avaliado pela ferramenta	68
4.11	Ferramenta gerada pela Engine	69
4.12	Dados referentes a uma avaliação inseridos	69
4.13	Interface para graduação de elementos do processo	70
4.14	Tela de cadastro de evidências	71
4.15	Botão para enviar a avaliação	71
4.16	Sumarização dos resultados da ferramenta	72
5.1	Metamodelo dos Dados de Entrada do Exemplo	74

5.2	Processo - Gerência de Projetos	75
5.3	Processo - Gerência de Requisitos	76
5.4	Planilha de Indicadores utilizada como exemplo comparativo	76
5.5	Planilha de Indicadores - Atributos de Processo	77
5.6	Ferramenta Gerada - Projetos	77
5.7	Ferramenta Gerada - Resultados e Atributos de Processo	78
5.8	Ferramenta Gerada - Tela de Cadastro de Evidências	78
5.9	Ferramenta Gerada - Tela de Sumarização dos Resultados	79
5.10	Experiência em Avaliação de Processos	82
5.11	Trabalho com Implantação de melhoria de procesos	82
5.12	Experiência com Ferramentas de Avaliação	82
5.13	Opinião sobre arquivos diferentes para uma única norma/modelo	83
5.14	Opinião sobre a interface da Engine	83
5.15	Opinião sobre a mecânica de geração da ferramenta	83
5.16	Opinião sobre organização dos dados na ferramenta	84
5.17	Opinião sobre o relacionamento de projetos da organização à avaliação	84
5.18	Opinião sobre o relacionamento de evidências aos projetos	84
5.19	Opinião sobre a graduação dos componentes do processo	85
5.20	Opinião sobre a persistência dos dados na ferramenta	85
5.21	Opinião sobre a sumarização dos dados na ferramenta	85
5.22	Opinião sobre o relatório emitido pela ferramenta	86

Lista de Tabelas

2.1	Mapeamento entre os níveis de maturidade do CMMI e do MPS.BR	28
2.2	Classes de Requisitos do Método de Avaliação do CMMI	31
4.1	Lista dos Requisitos Gerais	56
4.2	Lista dos Requisitos Específicos da Engine	57
4.3	Lista dos Requisitos Específicos da Ferramenta	58
4.4	Resumo dos artefatos de entrada da Engine	63
5.1	Comparativo entre as duas abordagens	80
5.2	Comparativo entre as ferramentas de avaliação	81

Sumário

1	Introdução	15
1.1	Motivação	15
1.2	Objetivos do Trabalho	17
1.3	Metodologia do Trabalho	18
1.4	Contribuições Esperadas	18
1.5	Estrutura do Documento	19
2	Fundamentação Teórica	20
2.1	Introdução	20
2.2	Normas de Qualidade	20
2.2.1	ISO/IEC 15504	20
2.2.2	CMMI-DEV	22
2.2.3	MPS-BR	26
2.3	Avaliação de processos de software	29
2.4	Métodos de Avaliação	30
2.4.1	SCAMPI	30
2.4.2	MA-MPS	35
2.5	Model-Driven Development	37
2.6	Requisitos Coletados	39
2.7	Considerações Finais	40
3	Trabalhos Relacionados	41
3.1	Introdução	41
3.2	ImPProS	41
3.3	ProEvaluator	44
3.4	ProEvaluator 2.0	46
3.5	FAPS	49
3.6	Requisitos Coletados	51
3.7	Considerações Finais	52
4	Processo de Desenvolvimento	53
4.1	Concepção	53
4.1.1	Objetivos	53
4.1.2	Requisitos	55
4.1.2.1	Requisitos Gerais	56
4.1.2.2	Requisitos Específicos da Engine	56

4.1.2.3	Requisitos Específicos das Ferramentas	56
4.2	Elaboração	57
4.2.1	Workflows	57
4.2.1.1	Workflow da Engine	58
4.2.1.2	Workflow das Ferramentas	59
4.2.2	Visão Geral e Arquitetura	60
4.2.3	Artefatos de Entrada	60
4.2.3.1	Metamodelo do Processo	61
4.2.3.2	Modelo do Processo	62
4.2.3.3	Método de Avaliação	62
4.2.4	Engine	63
4.2.4.1	Módulo de Leitura de Modelos	63
4.2.4.2	Módulo de Verificação de Modelos	63
4.2.4.3	Módulo de Geração de Classes	64
4.2.4.4	Módulo de Geração de Telas	64
4.2.4.5	Módulo de Geração de Ferramentas	64
4.2.5	Ferramentas de Avaliação	65
4.3	Avaliação	65
4.3.1	Exemplos de Uso	65
4.3.1.1	Entradas Utilizadas	65
4.3.1.2	Exemplo 1	66
4.3.1.3	Exemplo 2	68
4.4	Considerações Finais	72
5	Avaliação da Pesquisa	73
5.1	Introdução	73
5.2	Comparativo com Avaliação Real	73
5.2.1	Entrada	73
5.2.1.1	Processos	74
5.2.1.2	Atributos de Processo	75
5.2.2	Modelo dos Processos Avaliados	75
5.3	Planilha de Avaliação	75
5.3.1	Ferramenta Gerada	77
5.3.2	Comparação da Planilha de Avaliação com a Ferramenta Gerada	79
5.4	Comparativo com Outras Ferramentas	79
5.5	Avaliação por Usuários	79
5.5.1	Metodologia e Perfil dos Entrevistados	79
5.5.2	Resultados Obtidos	81
5.5.2.1	Questões Objetivas	81

5.5.2.2	Questões Subjetivas	86
5.5.3	Análise dos Resultados	87
5.6	Considerações Finais	87
6	Conclusão	88
6.1	Contexto	88
6.2	Contribuições do Trabalho	89
6.3	Trabalhos Futuros	90
	Referências	91
A	Apêndice A: Manual de Uso	94
B	Apêndice B: Questionário	102

1

Introdução

1.1 Motivação

Um processo de software é um conjunto de atividades, políticas, práticas e procedimentos, que são usados em uma empresa de software para desenvolver, implantar e manter um produto de software e os artefatos associados. E a medida de como esse processo é explicitamente definido, gerenciado, medido e executado é chamada de nível de maturidade (SUN et al., 2013).

O aumento do nível de maturidade de um processo de software dentro de uma organização é algo que é cada vez mais procurado e valorizado pelas empresas, figurando cada vez mais dentro do cotidiano de organizações de vários tamanhos (THIRY et al., 2008). Atividades como melhorias de processo e avaliação constante são apenas duas das etapas que auxiliam as organizações na busca constante por melhoria. Com o processo cada vez mais maduro, atividades de desenvolvimento vão se fortalecendo igualmente, pois o aumento da qualidade do processo e de suas atividades aumenta também a qualidade do produto final (CHEN; HOI; XIAO, 2011).

Como então medir a maturidade de um processo dentro de uma organização? Isso é feito através da avaliação do processo de software, que é uma tarefa conduzida dentro da organização para justamente medir o nível de maturidade do processo, para identificar e avaliar os seus pontos fortes e de melhoria, bem como áreas que melhoraram ou que precisam ser melhoradas. Baseado nestes resultados, a organização faz planos para a melhoria do processo visando atingir um nível de maturidade específico (SUN et al., 2013).

Outra questão que se faz presente é sobre qual o padrão ou norma que rege essa avaliação de processo de software. Outra questão abrange como saber se o processo da organização atingiu um nível de maturidade maior que o que ele possuía anteriormente. Para responder a essas questões, existem diversos modelos ou normas de qualidade. As mais conhecidas são o CMMI (SEI, 2010) e as ISO 15504 (ISO, 2004a; ISO, 2003; ISO, 2004b; ISO, 2004c; ISO, 2006). No caso do Brasil, existe ainda o MPS.BR (SOFTEX, 2013a). Estas normas atuam como modelos de processos, ou seja, definem o que uma organização precisa executar, em termos de processo para atingir os níveis de maturidade propostos.

Cada norma de qualidade possui um método de avaliação para aderência de processos à

norma, que segue um método previamente definido. Como exemplos de processos de avaliação, temos o SCAMPI (SEI, 2011), que avalia processos definidos de acordo com o CMMI, e o MA-MPS.BR (SOFTEX, 2013b), que avalia os processos de acordo com o MPS.BR. Cada um define um processo para a avaliação, com cada etapa contendo os artefatos a serem avaliados, bem como os resultados esperados.

A avaliação dos processos em geral é realizada através de evidências documentais, ou mesmo de afirmativas obtidas através de entrevistas (CHEN; HOI; XIAO, 2011). Aliado a isso, a avaliação dos processos de software de uma organização, frequentemente, é uma atividade puramente manual, em algumas ocasiões sendo auxiliada por planilhas eletrônicas.

Como são realizadas por especialistas em processos de software externos ao ambiente da organização (SUN et al., 2013), essa forma de avaliar pode, muitas vezes, introduzir uma subjetividade ao resultado da avaliação, que fatalmente poderia resultar em níveis diferentes para o mesmo processo, quando avaliado por pessoas diferentes.

Apesar de subjetiva, todo tipo de avaliação de processos de software segue um modelo ou norma. Elementos como nomenclatura, meios de avaliação, cronograma, critérios de avaliação, entre outros, são previamente definidos para auxiliar a avaliação. Tendo em vista essa uniformidade, despertou-se o interesse no desenvolvimento de formas de auxílio computacional mais efetivas do que planilhas ou documentos de texto virtuais.

A introdução dessas ferramentas visa auxiliar a realização das atividades que compõem os métodos de avaliação. Ainda que dependa da experiência e da forma de interpretar do avaliador, o uso de ferramentas leva a uma padronização relativa, que permite que avaliações realizadas por pessoas diferentes tenham menos diferenças em seus resultados.

Muitas vezes, uma empresa deseja ter seu processo avaliado de acordo com os critérios relativos a mais de uma norma de qualidade. Em grande parte dos casos, isso leva à utilização de mais de uma ferramenta, com diferentes tipos de usabilidade. Dependendo do número de normas a serem contempladas, essa adequação a diferentes cenários pode levar a algum transtorno ou confusão associada a esse chaveamento entre diferentes ferramentas e normas.

Diante desse cenário, uma das possíveis soluções para esse problema seria o desenvolvimento sob demanda de ferramentas customizáveis, onde o cliente escolheria que modelo ou norma a avaliação atenderia, e esta seria realizada. Porém, como haveria uma ferramenta para cada modelo ou norma, o custo total para a organização seria o de todas as ferramentas desenvolvidas. E a manutenção dessas ferramentas seria realizada a cada mudança no processo ou na norma, elevando ainda mais o custo.

Uma outra alternativa seria, baseado nos modelos de referência e de avaliação, criar ferramentas para avaliação dos processos. Nesse tipo de desenvolvimento, chamado de Model-Driven Development (MDD) (PASTOR et al., 2008), um ou mais modelos são fornecidos como entrada, e, após sucessivas transformações nos mesmos, é gerado código-fonte (ou outro modelo, dependendo do propósito).

Ou seja, a abordagem MDD permite que o desenvolvimento de uma ferramenta não

esteja mais baseado em uma norma específica desde o início do desenvolvimento, mas ao modelo ou norma que será utilizado para gerar a ferramenta para avaliar o processo. Isso torna a vida útil desses modelos mais longa, devido às revisões serem feitas nos mesmos, ou seja, no início do processo de desenvolvimento, pois os modelos estariam sob constante revisão e avaliação, para que estejam adequados à norma correspondente. Ou seja, isso evitaria a recodificação da ferramenta após várias iterações do processo de desenvolvimento, o que aumenta consideravelmente o custo (ATKINSON; KÜHNE, 2003). Assim, se uma nova versão do modelo for criada, seria necessário apenas transformar esse modelo em uma ferramenta de avaliação, seguindo as regras do MDD.

O uso de uma abordagem MDD se encaixa perfeitamente na necessidade de se ter um chaveamento eficiente entre vários contextos baseados em modelos. Ou seja, criando-se uma engine padrão para geração de ferramentas de avaliação, tudo se resume à questão de qual modelo de referência e avaliação seria usado, que é o propósito deste trabalho.

1.2 Objetivos do Trabalho

Este trabalho tem como objetivo apresentar uma engine para criação de ferramentas de avaliação de processos. Através da escolha de uma norma ou modelo de qualidade, a engine se baseará em modelos fornecidos como entrada para gerar automaticamente a ferramenta de avaliação correspondente.

Após o estudo de diversas ferramentas de avaliação disponíveis no mercado (OLIVEIRA; VASCONCELOS; ROUILLER, 2005; XAVIER, 2007; SOTERO, 2009; THIRY et al., 2008), os principais requisitos foram elicitados, e alguns outros foram adicionados para tornar a ferramenta mais acessível e simples de ser compreendida:

- Ser gerada a partir de modelos: Utilizar os conceitos de MDD para gerar as ferramentas de avaliação;
- Genérica para gerar diferentes ferramentas baseadas em diferentes modelos: Dado um modelo e seu método de avaliação (CMMI e SCAMPI, ou MPS-BR e MA-MPS), a ferramenta deve ser gerada baseada no modelo de entrada;
- Ser simples de executar, deixando a complexidade para a modelagem: Extrair o máximo possível de informações dos artefatos de entrada;
- Utilizar modelos MOF gerados pelo Eclipse Modeling Framework: O EMF foi escolhido com os objetivos de padronizar os modelos que vão ser recebidos como entrada, bem como ser um padrão para documentação dos mesmos.
- Requerer o mínimo possível de passos manuais para gerar a ferramenta: Uso intensivo de templates e bibliotecas visando agilizar a geração da ferramenta;

- Deve gerar uma ferramenta simples, de fácil entendimento e curva de aprendizagem baixa: Algo que possa ser preenchido com TABs, de maneira a agilizar o processo de preenchimento;

Após a definição desses requisitos, o resultado do trabalho visa atendê-los, de forma a criar uma ferramenta de fácil uso, e de utilidade prática para o avaliador do processo em questão. Ou seja, a usabilidade do ferramenta gerada pela engine deve ser boa o bastante para ser útil na avaliação real de um processo.

1.3 Metodologia do Trabalho

A metodologia do trabalho consistiu em duas fases distintas: a pesquisa acerca dos conceitos utilizados e a construção da solução em si. Embora tenham sido separadas, as duas fases foram realizadas de maneira concorrente.

Na pesquisa dos conceitos, algumas abordagens foram consideradas para modelagem dos modelos e normas de qualidade e seus métodos de avaliação. Nessa fase foi identificada que a utilização da abordagem MDD traria um ganho considerável para a construção da solução. Porém, as vantagens em futuras manutenções (dado o contexto de mudança de modelos) foram chave para essa escolha.

Outro tópico de pesquisa consistiu na procura por ferramentas que auxiliam na avaliação de processos de software, nas mais variadas plataformas. Estas ferramentas, juntamente com outros trabalhos realizados na área de automatização das atividades de avaliação de processos e os objetivos do trabalho consistiram na base do levantamento de requisitos da solução desenvolvida.

Com o embasamento conceitual concluído, assim como o conjunto de requisitos da solução, foi dado início ao desenvolvimento da solução. A plataforma, a linguagem de programação, o padrão de projeto foram escolhidos de maneira a atender os requisitos definidos, assim como os objetivos do trabalho. O software componente da solução proposta foi dividida em módulos, visando agrupar componentes coesos e facilitar mudanças no código.

A definição do padrão de modelos a ser utilizado como entrada no sistema consistiu de uma parte importante do desenvolvimento do trabalho. Esses modelos precisavam estar integrados com ferramentas de modelagem, visando facilitar sua criação por usuários.

Com o desenvolvimento concluído, a solução foi avaliada por profissionais que atuam na área de avaliação de processos de software. Isto se deu para que a solução fosse avaliada por usuários em contextos reais de avaliação, com diferentes normas e ferramentas.

1.4 Contribuições Esperadas

Com este trabalho, espera-se a criação de uma engine para geração de ferramentas de avaliação de processos de software baseada em modelos. Ou seja, de acordo com modelo,

criar uma ferramenta que atenda a requisitos básicos para a avaliação de software. Além disso, espera-se que seja flexível o bastante para atender a diferentes modelos e normas de qualidade, sem que sua interface seja alterada.

1.5 Estrutura do Documento

O trabalho aqui descrito tem, além deste capítulo introdutório, os seguintes capítulos:

- No Capítulo 2, a fundamentação teórica para o trabalho é apresentada, descrevendo os conceitos básicos para o entendimento do mesmo, como as normas e modelos de qualidade, bem como os seus métodos de avaliação associados.
- O Capítulo 3 descreve outras ferramentas e trabalhos relacionados a apoio à avaliação de maturidade processos de software.
- O Capítulo 4 descreve a engine em si, a estrutura geral das ferramentas a serem geradas, suas arquiteturas, requisitos e organização em geral, bem como exemplos de uso da engine e da ferramenta gerada.
- O Capítulo 5 apresenta uma análise crítica da ferramenta criada, através de um comparativo com os resultados de uma avaliação real, além da avaliação por profissionais que já participaram de avaliações e tiveram contato com ferramentas de avaliação.
- O Capítulo 6, é apresentada a conclusão do trabalho, bem como oportunidades de melhoria e trabalhos futuros.
- O Apêndice A, temos o manual de utilização da engine e das ferramentas, que foi utilizado para guiar a avaliação das mesmas.
- O Apêndice B, temos questões que foram respondidas pelos profissionais que foram entrevistados para avaliação deste trabalho.

2

Fundamentação Teórica

2.1 Introdução

Este capítulo visa apresentar em que conceitos teóricos este trabalho está baseado. Inicialmente são apresentadas as normas de qualidade que foram consideradas durante o desenvolvimento da solução. Na seção seguinte, o contexto de avaliações de processo de software é apresentado, relacionado às normas citadas. Então, na seção seguinte, os métodos de avaliação das normas são descritos. Em seguida, o MDD é explicado, apresentando os conceitos que foram utilizados para a construção da solução.

2.2 Normas de Qualidade

2.2.1 ISO/IEC 15504

Em 1993, A ISO realizou um estudo sobre as necessidades e os requisitos para avaliação de processos de software e concluiu que havia um consenso sobre a necessidade de um padrão universal único para avaliação de processos de software. O objetivo desse modelo era harmonizar os modelos já existentes e servir como referência para criação de modelos de avaliação de processos.

O resultado disso foi a criação do projeto SPICE (Software Process Improvement Capability dEtermination) (ROUT, 1995.) cujo objetivo era desenvolver uma norma internacional para avaliação de processos de software e realizar avaliações piloto antes da publicação como norma. que tinha como objetivo produzir inicialmente um relatório técnico que fosse mais geral e abrangente que os modelos existentes e mais específico que a norma ISO 9001 (NBR/ISO, 2000). Uma versão do SPICE foi aprovada em 1998 como Relatório Técnico e, apenas em 2003, a Norma ISO/IEC 15504 foi publicada, e foi dividida em partes (ISO, 2004a;ISO, 2003;ISO, 2004b) (ISO, 2004c;ISO, 2006). Porém, o modelo de avaliação de processos de software, que corresponde à parte 5 da norma, só foi lançado em março de 2006.

A norma ISO 15504 pode ser usada em dois contextos:

- **Melhoria Contínua:** identificando oportunidades de melhoria no processo da própria organização.
- **Determinação da Capacidade:** identificando o nível de capacidade dos processos dos fornecedores da organização.

Quando o objetivo da organização for a melhoria de processos, pode-se avaliá-los, gerando um perfil dos processos a ser utilizado na elaboração de um plano de melhorias. A análise dos resultados identifica os pontos fortes e fracos e os riscos inerentes aos processos. Já quando o objetivo da empresa for avaliar fornecedores para contratação, esta pode obter seus perfis de capacidade.

Seu modelo de referência define a dimensão de processo, que corresponde à definição de um conjunto de processos considerados universais e fundamentais para a boa prática da engenharia de software e a dimensão de capacidade, que corresponde à definição de um modelo de medição com base na identificação de um conjunto de atributos que permite determinar a capacidade de um processo para atingir seus propósitos, gerando os produtos de trabalho e os resultados estabelecidos.

Os processos do modelo de avaliação da ISO/IEC 15504 são descritos e classificados de maneira semelhante à norma ISO 12207 (ISO, 1995). A norma ISO/IEC 12207 tem como objetivo principal estabelecer uma estrutura comum para os processos de ciclo de vida de software visando ajudar as organizações a compreenderem todos os componentes presentes na aquisição e fornecimento de software e, assim, conseguirem firmar contratos e executarem projetos de forma mais eficaz. A norma ISO/IEC 15504 define 6 níveis de capacidade de processos sequenciais e cumulativos. Os níveis podem ser usados para avaliar como uma organização está realizando um determinado processo ou como um guia para a melhoria de processos.

Os níveis definidos pela norma são:

- **Nível 0 - Processo Incompleto:** há uma falha geral na satisfação do propósito do processo. Produtos de trabalho ou resultados de processos são poucos ou difíceis de serem identificados.
- **Nível 1 - Processo Executado:** o propósito do processo é geralmente alcançado, porém de uma forma não planejada e acompanhada. Existem produtos de trabalho e eles evidenciam a satisfação do propósito do processo.
- **Nível 2 - Processo Gerenciado:** o processo produz produtos de trabalho de acordo com procedimentos planejados e acompanhados. A execução do processo passa a construir produtos de trabalho que satisfazem os requisitos de qualidade especificados, dentro do cronograma de tempo e dos recursos necessários.
- **Nível 3 - Processo Estabelecido:** o processo utiliza um processo padrão que é capaz de atingir seus resultados definidos. A implantação de um processo usa uma versão customizada e aprovada de um processo padrão.

- **Nível 4 - Processo Previsível:** o processo passa a ser executado consistentemente dentro de limites definidos para atingir seus resultados. A qualidade dos produtos é conhecida de forma quantitativa.
- **Nível 5 - Processo Otimizando:** o processo definido e padrão é alterado e adaptado para atingir de forma efetiva os objetivos correntes e futuros do negócio. A otimização contínua do processo envolve experiências de idéias e tecnologias inovativas.

Na escala de avaliação dos atributos de processo definida pela norma ISO/IEC 15504, existem quatro resultados possíveis: totalmente, parcialmente, largamente ou não atingido. Cada nível contém um ou mais atributos de processo associados. O processo só atinge um determinado nível de capacidade se os resultados esperados dos seus atributos de processo correspondentes forem totalmente ou largamente atingidos. Fora isso, todos os atributos de processo dos níveis inferiores devem ter seus resultados avaliados como totalmente atingidos.

2.2.2 CMMI-DEV

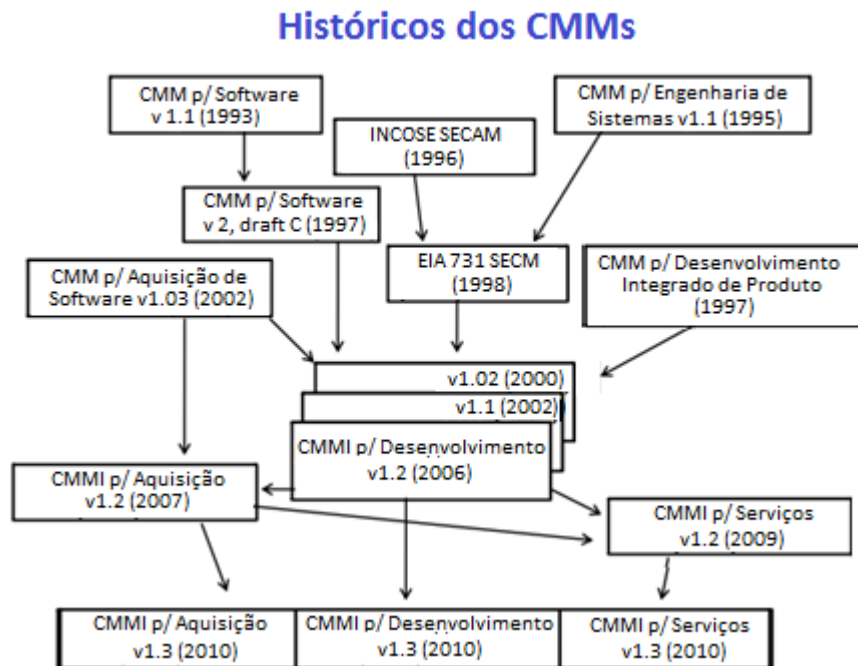
Criado em 2000, o CMMI (SEI, 2011) faz parte de um conjunto de modelos chamado CMM (Capability Maturity Models), cuja base de conhecimentos remonta aos anos 1990. Seu foco, basicamente, é melhorar processos em uma organização, descrevendo uma série de disciplinas que levam de um processo ad hoc a um processo maduro e organizado.

Neste contexto, o CMMI também provê guias para serem usados no desenvolvimento de processos. Não são processos propriamente ditos, nem descrições dos mesmos, pois isso seria impossível de ser feito dado a infinidade de contextos onde estes são aplicados. Não há como criar um processo-padrão, para que este possa ser usado em diferentes organizações de diferentes domínios de aplicação. Cada organização e cada domínio possui regras e padrões específicos, e uma padronização poderia inutilizar o processo para muitos contextos. Além do domínio de aplicação, vários outros fatores influenciam na criação do processo dentro de uma organização, que incluem o faturamento e tamanho da organização.

Os modelos de capacidade e maturidade evoluíram ao longo do tempo, como podemos ver na Figura 2.1.

O CMMI possui um framework que provê uma estrutura de componentes para modelagem, treinamento e avaliação de processos. Todos os modelos do CMMI possuem uma raiz comum dentro do framework, chamada CMMI Model Foundation (CMF). O CMF está presente em todos os modelos gerados, sendo combinados com as áreas específicas para produzir modelos. Essas áreas específicas são chamadas de constelações, sendo três: aquisição, desenvolvimento e serviços. Este trabalho foca mais na parte de desenvolvimento, conhecida como CMMI-DEV.

O CMMI-DEV (SEI, 2010) é um modelo de referência que cobre atividades de desenvolvimento tanto de produtos quanto de serviços. Suas práticas incluem gerenciamento de projetos, gerenciamento de processos, engenharia de sistemas, engenharia de hardware, entre

Figura 2.1: Evolução histórica do CMMI.

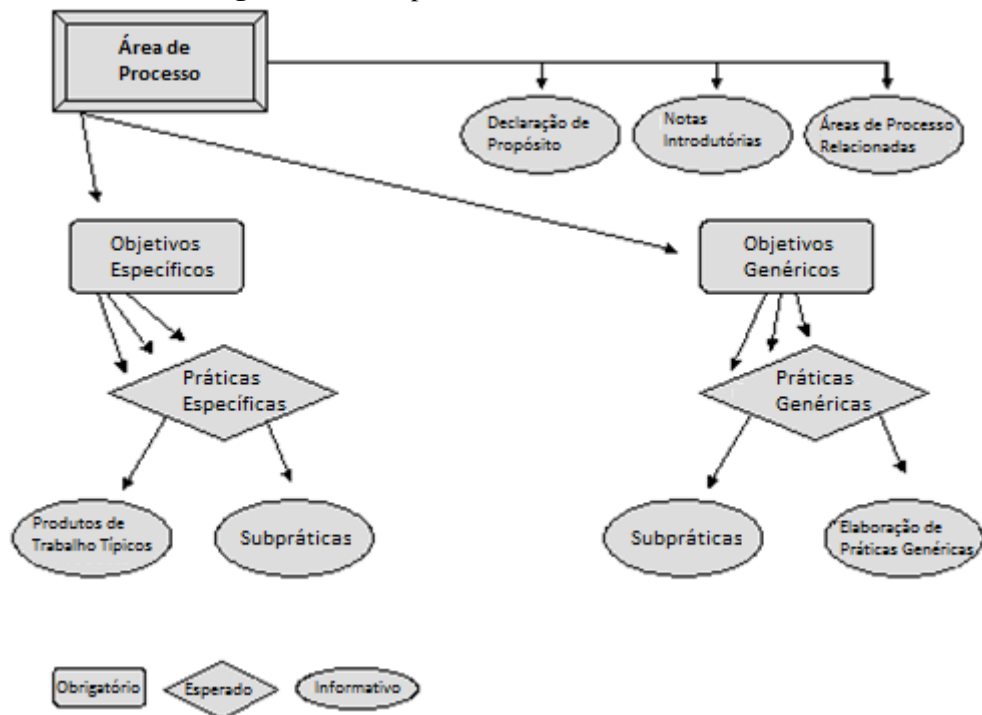
Fonte: O Autor

outros processos de suporte que fazem parte das atividades de desenvolvimento e manutenção dos produtos e/ou serviços da organização.

Organização

A organização do CMMI-DEV segue a organização do framework CMMI como um todo. Este contém todas as práticas e objetivos que são utilizados para produzir os modelos que pertencem às constelações, mais especificamente ao CMMI-DEV. Cada prática e objetivo pertence a uma categoria dentro do framework, que remetem ao nível de obrigatoriedade dos mesmos. Temos três categorias de componentes:

- **Componentes Obrigatórios:** são essenciais para uma dada área de processo, e devem ser visivelmente implementados no processo da organização. No framework, eles representam os objetivos específicos e genéricos.
- **Componentes Esperados:** que descrevem atividades que são importantes para se atingir um componente obrigatório do modelo. Eles guiam quem deve realizar as ações de melhoria ou de avaliação dos componentes obrigatórios.
- **Componentes Informativos:** dizem como devemos interpretar e entender componentes obrigatórios e esperados. Podem ser exemplos, subpráticas, referências, entre outros elementos auxiliares às atividades descritas nas outras categorias.

Figura 2.2: Componentes do modelo do CMMI.

Fonte: O Autor

A Figura 2.2 mostra como os elementos estão relacionados e a quais categorias pertencem dentro do CMMI. É importante entender os conceitos relacionados a cada elemento da Figura 2.2:

- **Process Areas (PA):** São elementos que agrupam práticas relacionadas a uma determinada área que, quando implementados de maneira correta, atingem um determinado conjunto de objetivos considerados importantes para melhoria dessa área.
- **Purpose Statements:** Descrevem o propósito da PA, e são componentes informativos do modelo.
- **Introductory Notes:** Descreve a maior parte dos conceitos cobertos pela PA, e são componentes informativos do modelo.
- **Related Process Areas:** Lista as PAs relacionadas à PA em questão, e reflete os relacionamentos em alto-nível entre elas. São componentes informativos do modelo.
- **Specific Goals (SG):** Descrevem as características únicas que devem estar presentes para satisfazer uma determinada PA. É um componente obrigatório no modelo e é usado em avaliações para auxiliar na determinação se uma área é satisfeita.
- **Generic Goals (GG):** São objetivos que são compartilhados por mais de uma PA. Eles descrevem características que devem estar presentes para institucionalizar processos que implementam uma PA. Assim como os SG, são utilizados em avaliações para

auxiliar na determinação se uma área é satisfeita. São componentes obrigatórios do modelo. São componentes informativos do modelo.

- **Specific Practices (SP):** São a descrição de uma atividade que é considerada importante para o SG ao qual ela está associada ser atingido. São componentes obrigatórios do modelo.
- **Typical Work Products:** Listam exemplos de saídas de uma SP. São componentes informativos do modelo.
- **Generic Practices (GP):** Semelhante ao conceito de GG, GPs são práticas que podem ser utilizadas por mais de uma PA. Eles descrevem características que devem estar presentes para institucionalizar processos que implementam uma PA.
- **Subpractices:** É uma descrição detalhada que provê orientações para interpretação e implementação de uma SP ou de uma GP. Seu propósito é fornecer ideias que podem ser úteis para melhorias no processo. São componentes informativos do modelo.
- **Generic Practice Elaborations:** Fornecem orientação em como GPs podem ser aplicadas unicamente para PAs. São componentes informativos do modelo.

Evolução de um processo no CMMI-DEV

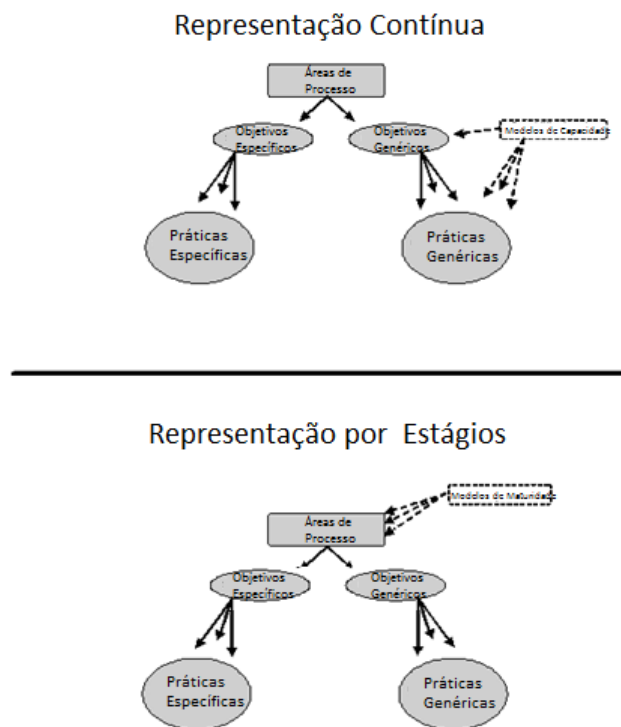
Depois de conhecer os elementos do modelo do CMMI, é preciso entender como eles são utilizados em conjunto, com o intuito de guiar uma organização na melhoria dos seus processos.

O CMMI não determina que a organização siga um processo pré-determinado, ou tenha que atingir objetivos específicos de performance (um número X de produtos por dia, por exemplo). Diferente disso, o modelo apenas especifica que a organização necessita de processos para direcionar práticas específicas de desenvolvimento.

Então, dentro do contexto do CMMI, processos são mapeados em áreas de processo, para permitir que a organização acompanhe o seu progresso de acordo com o modelo, por meio de atualizações e criação dos seus processos.

O acompanhamento do progresso é feito através de níveis. Os níveis no CMMI-DEV representam um caminho evolucionário recomendado a uma organização que deseja melhorar o processo que utiliza para desenvolver produtos ou serviços. O CMMI dá suporte a dois destes caminhos evolucionários, conhecidos também como representações: os níveis de maturidade e os níveis de capacidade.

Os níveis de capacidade habilita a organização a melhorar um conjunto de processos relacionados, abordando sucessivamente conjuntos de áreas de processo. Por isso, é chamada de representação contínua (ou Continuous Representation). Já os níveis de maturidade habilitam a organização a incrementalmente melhorar processos correspondentes a uma área de processo

Figura 2.3: Estrutura das Representações.

Fonte: O Autor

individual (ou grupos de áreas de processo), selecionadas pela organização. Devido a isso, é chamada de representação por estágios (ou Staged Representation).

É importante frisar que, apesar de serem dois caminhos distintos, ambas são importantes para melhoria dos processos, e ambas podem ser aplicadas aos mesmos componentes do modelo. A Figura 2.3 exibe uma comparação gráfica entre os dois caminhos.

Para atingir um nível, a organização precisa satisfazer todos os objetivos da área de processo (ou um conjunto delas) que são alvo da melhoria, independente de ser nível de capacidade ou de maturidade.

2.2.3 MPS-BR

Apesar de existirem diversos modelos e normas de qualidade, ainda existem organizações que não podem arcar com os custos da implantação de processos, nem de adequação de um processo já existente a padrões internacionais. Isso representa uma perda significativa de competitividade de empresas que muitas vezes estão começando as suas operações no mercado. Essa situação é uma realidade preocupante, principalmente no cenário brasileiro (SOFTEX, 2013a).

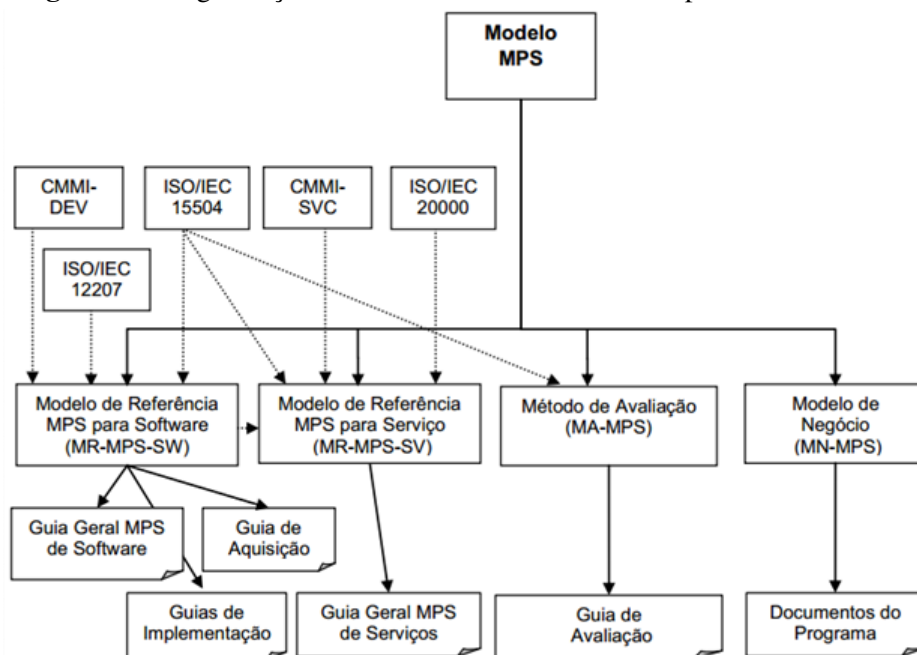
O modelo de Melhoria de Processo do Software Brasileiro (MPS-BR) surgiu para atender demandas de organizações de vários tamanhos e características, embora tenha um foco especialmente voltado para micro, pequenas e médias empresas. Apesar de ser uma iniciativa

brasileira, espera-se que o modelo atenda a modelos de qualidade de padrão internacional, visando não perder a compatibilidade com iniciativas já existentes.

Para atingir esses objetivos, o modelo visa que empresas tenham em foco melhoria de processos, através da utilização de princípios da engenharia de software. Ou seja, espera-se que a qualidade dos produtos de software seja aumentada através da melhoria dos processos de desenvolvimento dos mesmos.

Semelhante ao que ocorre no CMMI, o modelo MPS-BR baseia-se em níveis de capacidade e de maturidade para avaliação e melhoria da qualidade e produtividade de software e serviços, embora seja avaliado apenas de forma estagiada. O modelo MPS como um todo possui quatro componentes: o Guia Geral de Software (MR-MPS-SW) (SOFTEX, 2013b), Guia Geral de Serviços (MR-MPS-SV) (SOFTEX, 2013c), Guia de Avaliação (MA-MPS) (SOFTEX, 2013b) e Guia de Aquisição (SOFTEX, 2013d). Nesta seção, focaremos no MR-MPS-SW. O MA-MPS vai ser explicado na seção 2.4. A organização do modelo MPS, com seus componentes e os respectivos guias e documentos está descrita na Figura 2.4.

Figura 2.4: Organização do Modelo MPS com seus componentes e artefatos.



Fonte: SOFTEX, 2013a

O modelo do MPS.BR para Software define níveis de maturidade, que são uma combinação de processos e sua capacidade. A definição desses processos é feita declarando o propósito e os resultados esperados do mesmo, como acontece na ISO/IEC 15504-2 (ISO, 2004b). As atividades e tarefas necessárias para atender aos requisitos dos processos definidos fica a cargo de quem vai implementar, estando fora do escopo do modelo.

Já a capacidade de um processo é sua habilidade de atender os objetivos do negócio, estando relacionada com a satisfação de cada um dos atributos do processo em cada nível de maturidade do modelo.

Níveis de Maturidade

Os níveis de maturidade medem o quanto um processo evoluiu, em termos de melhorias, dentro de uma organização. Esses níveis permitem que previsões de comportamento de um ou mais processos sejam feitas, ajudando a organização na execução dos mesmos. O MPS.BR-SW estabelece sete níveis de maturidade: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). A escala inicia no G e tem no A seu nível mais alto. A cada um desses níveis é atribuído um perfil de processos que indicam em que áreas uma organização deve focar o esforço de melhoria.

Tabela 2.1: Mapeamento entre os níveis de maturidade do CMMI e do MPS.BR

CMMI	MPS.BR
1	Não é definido
2	G
2	F
3	E
3	D
3	C
4	B
5	A

A Tabela 2.1 compara os níveis de maturidade do CMMI com os do MPS.BR. A divisão em 7 níveis aumenta a chance de micro, pequenas e médias empresas implementarem e avaliarem os seus processos. Um dos fatores que comprovam isso é que mais níveis habilitam que melhorias nos processos sejam vistas em prazos mais curtos, o que teoricamente, é mais adequado à realidade financeira das empresas emergentes (SEI, 2010).

Processo

O processo, dentro do MPS.BR-SW, é descrito em termos de propósito e resultados. O propósito descreve o objetivo a ser atingido pelo processo durante sua execução. Os resultados descrevem o que se espera obter com a implementação efetiva do processo, que pode representar um produto ou mudança significativa no estado do mesmo.

Capacidade de Processo

É representada por um conjunto de atributos de processo (AP), descritos em termos de resultados esperados. A capacidade de processo representa o quão refinado e institucionalizado

o processo está no contexto organizacional. A evolução nos níveis de maturidade reflete diretamente no aumento da capacidade do processo da empresa.

Os atributos de processo são atendidos pela obtenção de resultados esperados do atributo de processo (RAP). A obtenção desses resultados se dá de maneira acumulativa entre os níveis de maturidade do processo. Por exemplo, estando no nível E, o processo precisa também atender aos APs dos processos de níveis F e G, embora os processos anteriormente implementados sejam executados todos no nível superior de maturidade. Pois à medida em que um projeto vai crescendo em maturidade, espera-se que o que já foi implementado também se desenvolva, e de forma contínua. Um AP pode ter diferentes RAPs para diferentes níveis de maturidade.

2.3 Avaliação de processos de software

Uma avaliação de processos de software é uma tarefa conduzida por uma organização com o objetivo de medir a maturidade de um processo de software, visando identificar e avaliar as fraquezas, forças, e pontos a melhorar do processo (SUN et al., 2013). Mas para avaliar o processo precisamos de parâmetros para o mesmo, ou seja, poder comparar o processo implementado com algum outro, previamente criado e validado e que possua sua maturidade reconhecida.

Um modelo de processo define abstratamente que elementos um processo precisa conter para que este resulte em um produto de qualidade em seu final, respeitando sempre as características da organização na qual o processo vai ser inserido (XAVIER, 2007). O objetivo principal de uma avaliação de processo, consequentemente, é verificar e medir o quão bem este processo está definido de acordo com o modelo escolhido.

A atividade de avaliação de processos cresce cada vez mais em importância dentro das organizações, visto que a qualidade do processo está cada vez mais ligada à qualidade do produto final. Garantindo a qualidade do processo como um todo, a qualidade do produto é uma consequência. Por isso a incorporação cada vez maior dentro das atividades do processo de software da organização.

Essas avaliações normalmente são feitas baseadas em entrevistas com membros da equipe do projeto, ou ainda, checando aleatoriamente artefatos gerados pelo processo (por exemplo, verificando códigos armazenados em um sistema de gerenciamento de configuração), entre outros meios (CHEN; HOI; XIAO, 2011). Esses métodos geralmente exigem uma grande experiência em avaliações de processo por parte de quem vai conduzir a atividade, pois lida com processos de vários domínios e precisa definir várias questões para conduzir as avaliações de acordo com o ambiente organizacional. Consequentemente, demanda um custo consideravelmente alto.

Para empresas de pequeno ou médio porte, ou ainda, organizações que não possuem ainda um processo definido e/ou bem fundamentado, o custo tem grandes chances de ser proibitivo, a princípio. Isso é um desastre para empresas que querem conquistar um lugar num mercado competitivo como o de software.

Ainda neste contexto, observamos que a organização precisa melhorar sempre a qualidade

dos seus produtos, o que leva a um monitoramento constante desta qualidade. Ou seja, a atividade de avaliar, já custosa de início precisa ser repetida várias vezes. Por isso, está sendo cada vez mais procurado o apoio de ferramentas visando agilizar e automatizar essa avaliação dos processos de uma organização, visando a diminuição do custo associado (CHEN; HOI; XIAO, 2011).

Com isso, várias abordagens surgiram para tentar minimizar a dependência da expertise de um avaliador (ou grupo de avaliadores) na atividade. Obviamente, a interpretação dos resultados é ainda bastante subjetiva, mas as implementações se mostraram eficientes em auxiliar a avaliação, minimizando os erros e diminuindo o tempo de compilação dos resultados (CHEN; HOI; XIAO, 2011).

Com a expansão dos mercados e a necessidade de competir com empresas internacionais, empresas estão buscando estar de acordo com normas de modelos de abrangência internacional. Como o objetivo dessas abordagens é auxiliar as empresas nessa tarefa, elas em sua grande maioria são baseadas nos modelos e normas mais bem aceitos pela comunidade brasileira e internacional: a norma ISO 15504, o CMMI e o MPS.BR, que foram detalhados na seção anterior.

2.4 Métodos de Avaliação

2.4.1 SCAMPI

Antes de definir o método de avaliação SCAMPI (SEI, 2011), componente do CMMI, é preciso explicar como o CMMI organiza os métodos de avaliação para processos: através do *Appraisal Requirements for CMMI*, ou ARC (SEI, 2011).

ARC

A estrutura de classe do método de avaliação do CMMI identifica os requisitos apropriados para métodos de avaliação desenvolvidos especificamente para três aplicações típicas, mostrado na Tabela 2.2 Não há nenhuma exigência para qualquer método de avaliação a estar dentro de uma determinada classe; no entanto, esta estrutura se destina a fornecer valor e utilidade para os utilizadores do CMMI, e seu uso é incentivado.

Atributos diferenciadores chave para as classes de avaliação incluem o seguinte:

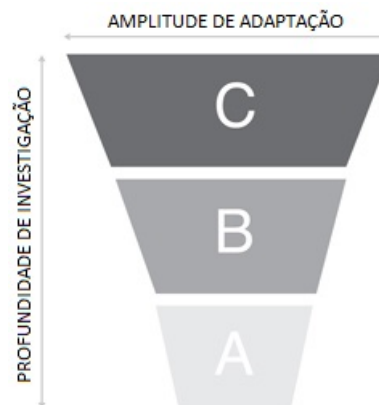
- O grau de confiança nos resultados de avaliação;
- A geração de classificações;
- Custo de avaliação e duração;

Um comparativo gráfico entre as classes pode ser visto na Figura 2.5

Métodos da classe A devem satisfazer todos os requisitos ARC e são os únicos métodos considerados adequados para a prestação de classificação para benchmarking. O SEI vai apenas

Tabela 2.2: Classes de Requisitos do Método de Avaliação do CMMI

Requisitos	Classe A	Classe B	Classe C
Tipos de evidências objetivas recolhidas	Artefatos e afirmações	Artefatos e afirmações	Artefatos e / ou afirmações
Pontuações Geradas	Classificações de Objetivos Obrigatória	Não permitido	Não permitido
Unidade Organizacional	Cobertura Obrigatória	Desnecessária	Desnecessária
Requisitos de Lídes de Avaliação	Avaliador Certificado	Pessoa treinada e experiente	Pessoa treinada e experiente

Figura 2.5: Comparação entre as classes ARC.

Fonte: SOFTEX, 2013a

reconhecer avaliações geradas por métodos SCAMPI de classe A, embora possam ser criados outros métodos baseados na classe A do ARC.

Métodos de avaliação de classe B devem estar em conformidade com algum subconjunto de requisitos ARC. Vários requisitos de métodos de classe A são opcionais para os métodos de classe B. Dois tipos de provas objetivas são necessárias tanto para métodos das classes A e B. Métodos de classe B não produzem classificações. Estes tipos de avaliações são recomendados para métodos iniciais em organizações que estão apenas começando a usar modelos CMMI para atividades de melhoria de processo, pois fornecem um meio de baixo custo para a realização de avaliações anteriores à avaliações por métodos de classe A.

Já métodos de avaliação da classe C atuam como um subconjunto de requisitos ARC de métodos de classe B. Apenas um dos dois tipos de provas objetivas necessárias para métodos da Classe A e Classe B são exigidos para os métodos de classe C. Métodos de avaliação de classe C não produzem classificações, sendo sua validação e corroboração opcionais. Este tipo de avaliação seria mais adequado para casos onde há apenas a necessidade de uma "olhada rápida" em um processo, ou para auto-avaliações periódicas feitas por projetos ou grupos de

apoio organizacional.

Requisitos ARC são baseados em métodos de avaliação largamente utilizados, que renderam resultados consistentes, precisos e úteis. Como outros métodos de avaliação vão sendo identificadas e podem possuir características semelhantes, os requisitos podem ser modificados para absorver estas novas características.

Definição

O SCAMPI (Standard CMMI Appraisal Method for Process Improvement) foi criado como medidas de referência de qualidade para modelos baseados no CMMI. Embora tenha sido projetado para modelos CMMI, pode ser também aplicado para avaliações de outros modelos de referência do CMM. Na prática, um método de avaliação, como o SCAMPI, habilita uma organização que quiser comparar suas realizações de melhoria de processos com outras organizações do setor, e, além disso, atingir um determinado nível de maturidade como parte da avaliação do processo.

O método SCAMPI permite à organização:

- Ter uma visão sobre a capacidade de uma organização através da identificação dos pontos fortes e fracos seus processos atuais em relação ao modelo de referência;
- Priorizar planos de melhoria;
- Identificação e foco em melhorias mais benéficas para a organização como um todo, dado o atual nível de maturidade da mesma ou de maturidade de seus processos;
- Derivar classificações de níveis de capacidade, assim como de níveis de maturidade
- Identificar riscos relativos às determinações de capacidade e de maturidade.

As decisões tomadas com base em avaliações de nível de maturidade só são válidas se as classificações são baseadas em critérios conhecidos. Consequentemente, a informação contextual, organizacional da unidade, o escopo organizacional, o escopo do modelo de referência, tipo de método de avaliação, a identidade do líder da equipe de avaliação e a equipe são itens para os quais os critérios e as orientações são fornecidas dentro do método para garantir uma interpretação coerente dentro da comunidade. O benchmarking só pode ser válido quando há um base consistente para estabelecer os pontos de referência.

O SEI mantém dados agregados da indústria, com os resultados de avaliação recolhidos a partir de organizações que realizaram avaliações desde 1987. O perfil dessa indústria é baseado em dados de avaliação prestados por profissionais treinados e reconhecidos pelo SEI, e é atualizado duas vezes por ano.

Conceitos Básicos e Abordagem

SCAMPI A, como um método de aferição de avaliação, depende de uma agregação de informação que é recolhidas através tipos definidos de provas objetivas. A prova objetiva alimenta um "information-processing motor", cujas partes são constituídos por uma série de transformações de dados. A avaliação equipe observa, ouve e lê a informação que se transforma em notas, e em seguida, em caracterizações de lacunas de implementação de prática ou de conformidade, e em seguida, em resultados preliminares. Estes resultados são validados pela organização antes de se tornarem resultados definitivos. O conceito fundamental é que estas transformações são aplicadas aos dados que refletem a processos promulgados na unidade organizacional e do modelo de referência de avaliação, e este conjunto de dados é a base para as classificações e de outros resultados de avaliação.

Planejamento é fundamental para a execução do SCAMPI A. Todas as fases e atividades do processo brevemente discutidas abaixo derivam de um plano bem articulado desenvolvido pelo líder da equipe de avaliação em conjunto com os membros da organização avaliada e o patrocinador da avaliação.

Metodologia

O SCAMPI A contém três fases, que serão detalhadas a seguir.

Fase 1: Planejar e preparar para a avaliação

O planejamento da avaliação começa com a compreensão do objetivos, as exigências da organização, além de restrições. Todas as outras fases, planejamento, preparação, execução e apresentação dos resultados derivam a partir desta atividade inicial. Devido ao significativo investimento e planejamento logístico envolvido, essa fase demanda bastante cuidado em sua realização. Como em cada fase subsequente, o valor da iteração diminuirá à medida que os dados são coletados, analisados, refinados, e traduzidos em resultados de importância relativa para o modelo. Ou seja, menos esforços e recursos vão ser direcionados para a obtenção dos dados para a avaliação.

Uma equipe de profissionais experientes e qualificados executa uma avaliação SCAMPI ao longo de um período de tempo negociado entre a organização e o líder da equipe de avaliação. O escopo da organização para ser avaliado, bem como o escopo do modelo de referência de avaliação (áreas de processo), tem de ser também definido. O escopo do modelo e da organização fornecem a base para estimar os custos de tempo, de pessoal, logísticos (viagens, por exemplo), além dos custos globais para a organização avaliada.

Durante a avaliação, a equipe de avaliação verifica e valida cada prova objetiva fornecida pela organização avaliada para identificar pontos fortes e fracos em relação à referência do

modelo de avaliação. A evidência objetiva é composta de artefatos e afirmações utilizadas como indicadores para implementação e institucionalização das práticas do modelo. Antes da fase de condução da avaliação começar, os membros da organização avaliada normalmente coletam e organizam documentos como prova objetiva, por meio de estratégias de coleta de dados definidos com base nos artefatos disponíveis dentro da organização e alinhado com o modelo de referência de avaliação.

Preparação prévia, tanto pela equipe de avaliação quanto da organização avaliada é a chave para a execução mais eficiente do método. Análise preliminar de evidências objetivas documentadas fornecidas pela organização avaliada tem um papel importante na preparação para a execução da avaliação. Se dados substanciais deixarem de ser coletados neste momento, as atividades de avaliação subsequentes podem ser adiados ou mesmo cancelados, pois os dados coletados podem ser julgados insuficientes para compensar a atividade global de avaliação.

A coleta de evidências documentadas, em algum grau, pela organização avaliada antes da avaliação pode ajudar a melhorar a eficiência da equipe de avaliação, mas também pode oferecer vários outros benefícios para a organização:

- Melhoria da precisão nos resultados de avaliação entregues por equipes de avaliação externa (ou seja, clara compreensão dos processos implementados, pontos fortes e pontos fracos);
- Obtenção de uma compreensão detalhada de como cada parte da organização participante da avaliação implementou práticas do modelo, bem como o grau de cumprimento e de adaptação aos processos organizacionais padrões;
- Estabelecer os artefatos gerados pela avaliação que podem ser reutilizados em avaliações posteriores, minimizando o esforço necessário para a preparação de uma avaliação futura.

No entanto, o esforço de coletar, organizar e rever grandes quantidades de evidências objetivas antes da avaliação pode ser um grande custo para as organizações avaliadas, e pode levar a grandes impactos no orçamento das mesmas se não for feito de forma eficiente. Estratégias de coleta de dados incrementais com solicitações de dados específicos pode ajudar a mitigar os riscos, como uso recursos da organização de forma ineficiente em coleta de dados, ou ainda que resultem em dados que não são apropriados ou úteis. Um plano de coleta de dados, desenvolvido pelo líder da equipe de avaliação em conjunto com a organização avaliada, pode ajudar a tornar explícito quanto esforço vai ser necessário para coleta de dados, bem como ele será distribuído entre a organização e a equipe de avaliação.

Fase 2: Condução da avaliação

Na fase 2, a equipe de avaliação centra-se na coleta de dados da organização avaliada para julgar o quanto do modelo está implementado. Integrado com esta abordagem é o conceito de cobertura, o que implica duas coisas: em primeiro lugar, o recolhimento de dados suficientes para cada componente no âmbito do modelo escolhido pelo patrocinador e, segundo, a obtenção de uma amostra representante dos processos em curso.

A amostragem é planejado quantitativamente, e é baseada na diversidade de implementações únicas do processo no âmbito de avaliação, com o objectivo de assegurar tanto uma amostra representativa da unidade organizacional, quanto uma otimização no esforço para a coleta e análise de evidências objetivas. Isso significa a coleta de dados e informações sobre todas as práticas do modelo de referência de avaliação do escopo de avaliação, através de instâncias de processos amostrados dentro da unidade organizacional a ser avaliada. O plano de coleta de dados desenvolvido na fase 1 sofre iteração contínua e refinamento até que a cobertura suficiente seja atingida.

Ao determinar que a cobertura suficiente do modelo de referência de avaliação e organizacional unidade foi obtida, os resultados de avaliação podem ser gerados. Classificações para objetivos são determinadas dentro de cada área de processo, que coletivamente podem ser usados para determinar a classificação aggrada para as áreas de processo individuais ou para a unidade organizacional, conforme o caso.

Fase 3: Relatório de Resultados

Na fase 3, a equipe de avaliação fornece os resultados e as classificações para o patrocinador da avaliação e da organização. Esses artefatos tornam-se parte do registro de avaliação, que se torna dados protegidos em acordo com a declaração de divulgação de avaliação. O nível de protecção e do plano de disposição de materiais de avaliação e os dados são determinados na fase 1, em colaboração com o patrocinar. Um pacote de dados completado a avaliação, que inclui um subconjunto de o conteúdo do registro de avaliação, está prevista para o SEI. O SEI adiciona os dados de avaliação de bancos de dados confidenciais, e fornece perfis globais da comunidade em uma base periódica.

2.4.2 MA-MPS

Foi criado com o objetivo de descrever o Método de Avaliação MA-MPS (SOFTEX, 2013b), além dos requisitos para avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IA).

O Método de Avaliação MA-MPS foram definidos de forma a:

- permitir a avaliação objetiva dos processos de software e de serviços de uma organização/unidade organizacional;

- permitir a atribuição de um nível de maturidade do MR-MPS-SW e MR-MPS-SV com base no resultado da avaliação;
- ser aplicável a qualquer domínio na indústria de software;
- ser aplicável a organizações/unidades organizacionais de qualquer tamanho.

O propósito do Método de Avaliação MA-MPS é verificar a maturidade da unidade organizacional na execução de seus processos de software e de serviços.

O processo de avaliação descreve o conjunto de atividades e tarefas a serem realizadas para atingir este propósito. Ele tem início com a seleção de uma Instituição Avaliadora (IA) e encerra com o registro dessa avaliação na base de dados confidencial da SOFTEX.

O patrocinador pode ser um representante da alta gerência da unidade organizacional a ser avaliada, ou de uma outra organização que solicita a avaliação da unidade organizacional por uma terceira parte para fins de contrato. Para que uma avaliação seja conduzida com sucesso, é necessário:

- Comprometimento do patrocinador: o comprometimento do patrocinador é essencial para assegurar que os objetivos da avaliação sejam atingidos. Este comprometimento também diz respeito aos recursos necessários, tempo e pessoal disponível para executar a avaliação.
- Motivação: a atitude da gerência da unidade organizacional tem forte impacto nos resultados de uma avaliação. O responsável pela unidade organizacional deve motivar os participantes de forma aberta e construtiva. Deve, também, deixar claro a todos que o foco da avaliação é o processo e não o desempenho dos indivíduos que implementam o processo.
- Fornecimento de feedback: o fornecimento de feedback e o estabelecimento de uma atmosfera que encoraje a discussão aberta sobre os resultados preliminares, durante a avaliação, ajudam a assegurar que a avaliação seja significativa para a unidade organizacional.
- Confidencialidade: o respeito à confidencialidade das informações obtidas durante a avaliação é essencial para que se obtenha as informações necessárias. Deve-se assegurar total confidencialidade aos participantes, tanto da equipe de avaliação quanto dos entrevistados. Da mesma forma, esse entendimento sobre a confidencialidade é essencial para que nenhum entrevistado se sinta ameaçado e todos se expressem livremente. A confidencialidade também abrange as informações contidas nos documentos apresentados pela unidade organizacional aos avaliadores.
- Percepção dos benefícios: os membros da unidade organizacional devem perceber que a avaliação resultará em benefícios que os ajudarão direta ou indiretamente a realizar o seu trabalho.

- **Credibilidade:** o patrocinador, o gerente e os colaboradores da unidade organizacional devem acreditar que a avaliação chegará a um resultado representativo da organização/unidade organizacional. É importante que todas as partes confiem que os avaliadores têm a experiência e competência para realizar a avaliação, são imparciais e têm um entendimento adequado da unidade organizacional.

Uma avaliação seguindo o MA-MPS tem validade de 3 (três) anos a contar da data em que a avaliação final foi concluída na unidade organizacional avaliada.

2.5 Model-Driven Development

Um dos grandes desafios da Engenharia de Software atualmente é aumentar cada vez mais os níveis de abstração com que engenheiros de software escrevem programas. Neste contexto, o desenvolvimento de software a partir de modelos é uma continuação natural dessa tendência (ATKINSON; KÜHNE, 2003).

A atividade de modelagem de elementos na Engenharia de Software vem crescendo de importância com o decorrer dos anos. Com isso, um grande número de abordagens para essa modelagem tem surgido, muitas dessas informais. Linguagens para modelagem surgiram, bem como padrões de diagramas para descrição, especificação de sistemas e de relacionamento dos dados nos mesmos (MOHAGHEGHI; AAGEDAL, 2007).

Utilizados inicialmente para a comunicação entre stakeholders ou para fornecer uma representação gráfica de um sistema, os modelos passaram a oferecer outras utilidades para a equipe responsável por um sistema. O código-fonte do software que resultaria do processo de desenvolvimento passou a ser resultado de transformações sucessivas em modelos. (VASCONCELOS et al., 2011).

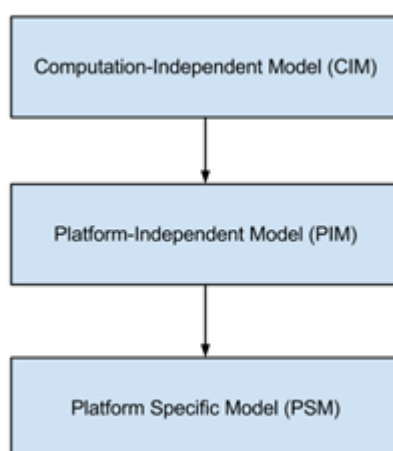
Desenvolvimento orientado a modelos (ou Model-Driven Development - MDD) compreende na aplicação de transformações em modelos durante um processo de desenvolvimento (PASTOR et al., 2008). Isto se dá de maneira (semi)automática, e de um nível mais abstrato para um nível mais concreto, ou ainda, do mesmo nível de abstração.

Várias abordagens e paradigmas foram criados para a aplicação de um processo MDD (ou ainda MDE, de Model-Driven Engineering). O principal paradigma é o MDA (Model-Driven Architecture), proposto pela OMG (OMG, 2013). Essa abordagem divide os modelos em três níveis de abstração diferentes, como mostrado na Figura 2.6:

Modelo Independente de Computação

Representa o primeiro modelo do processo de desenvolvimento. Tem seu foco no ambiente em que o sistema vai ser inserido, bem como de seus requisitos iniciais. Além disso, como o próprio título descreve, é completamente independente de qualquer computação. É referenciado através da sigla CIM (Computation-Independent Model)

Figura 2.6: Relacionamento entre os modelos do MDD.



Fonte: O Autor

Modelo Independente de Plataforma

Como modelo seguinte, temos o modelo PIM (Platform-Independent Model), que leva em consideração que partes do sistema vão ser efetivamente desenvolvidas como software, mas ainda não determina que tecnologia será utilizada para transformar o modelo no produto final do processo.

Modelo Específico de Plataforma

Depois de definidas as partes computáveis do sistema, é preciso determinar que tecnologia (no caso, a plataforma que em que o sistema vai ser baseado). Esta é a função do modelo específico de plataforma, ou PSM (Platform Specific Model).

A partir do PSM, então, o modelo de código (ou código-fonte) do sistema é gerado. Além disso, quanto mais subjetiva for a modelagem, mais complicada de ser definida ela se torna, pois ela passa a depender de quem vai utilizar o modelo gerado. Dentro de um processo MDD, os mais variados modelos gerados estão classificados entre os três tipos principais (SIM, PIM ou PSM).

Múltiplos modelos auxiliam no gerenciamento e descrição de um sistema complexo, onde cada um desses modelos representa um aspecto diferente da solução a ser desenvolvida (SENDALL; KOZACZYNSKI, 2003). Trabalhando com estes modelos, e sendo estes interligados, o esforço para garantir a consistência de todos eles é consideravelmente alto. Além de diminuir o tempo para sincronização horizontal (entre modelos do mesmo nível de abstração) e

vertical (modelos de diferentes níveis de abstração), a automação de transformações de modelos pode reduzir o peso de atividades como engenharia reversa, aplicação de padrões, refatoramento, etc.

Com isso, cresceu a importância dos modelos durante todo o ciclo de vida de um processo de desenvolvimento. Cada vez mais, a qualidade de um sistema depende da qualidade do modelo que o especifica. Devido a isso, a atividade de modelar um sistema não está restrita apenas aos diagramas de funcionalidades, que representam a parte funcional do sistema. Também abrange os modelos de qualidade que serão considerados durante o desenvolvimento, linguagens de modelagem, ferramentas, processos de modelagem e as transformações feitas nos modelos ao longo do processo (MOHAGHEGHI; DEHLEN; NEPLE, 2009).

Com o crescente uso de abordagens para desenvolvimento orientado a modelo (ou Model-Driven Development - MDD), encontrar maneiras de relacionar modelos de qualidade ao processo de desenvolvimento se tornou um desafio cada vez maior. Pois o produto em si tem seu desenvolvimento iniciado nas fases iniciais do processo, por meio de modelos sendo então transformado para código executável ao longo do processo.

Então, pode-se afirmar que relacionar qualidade ao processo de modelagem tem o mesmo resultado de trabalhar a qualidade do produto resultante do processo, dado um contexto MDD (PASTOR et al., 2008). Portanto, entender o funcionamento do software através de transformações sucessivas em um modelo contribuem para aumentar a sua qualidade (GITZEL; KORTHAUS, 2004). Pois pelo acompanhamento da evolução do modelo é possível obter uma visão geral de como os componentes do software interagem entre si, e possivelmente como interagem com os stakeholders. Com isso, várias abordagens surgiram para relacionar modelos de qualidade com ambientes MDD foram desenvolvidas (MOHAGHEGHI; DEHLEN, 2008), para a qualidade ser considerada desde a modelagem inicial do processo.

2.6 Requisitos Coletados

A revisão da literatura gerou um conjunto de requisitos com foco no processo de desenvolvimento da solução, assim como a análise das normas e modelos de qualidade. Estes, bem como seus métodos de avaliação precisaram ser avaliados quanto à viabilidade de uso na abordagem MDD.

Os requisitos coletados dizem respeito principalmente à modelagem dos processos. Isso permitiu a montagem dos artefatos de entrada da solução proposta por este trabalho. O estudo e entendimento claro dos vários modelos e normas de qualidade descritos foi fundamental para a criação da estrutura sobre a qual a solução foi contruída.

2.7 Considerações Finais

A ideia de automatizar uma avaliação de software não é necessariamente nova. Ainda assim, são encontrados várias oportunidades de melhoria nos meios automáticos de avaliação de processos, tais como adequação a atualizações nas normas de qualidade e criação baseada diretamente nos modelos ou normas de qualidade. Existem, porém, obstáculos a essa busca por uma forma automática eficiente de auxílio a avaliações.

O MDD surge como uma alternativa que visa dar mais agilidade ao processo de desenvolvimento de soluções que automatizem tanto o processo de avaliação, quanto a análise de resultados da mesma. O uso de modelos, como matéria-prima para geração de código diminui em muito o custo de desenvolvimento de ferramentas distintas para modelos de referências e modelos de avaliação distintos.

Esse cenário se encaixa perfeitamente na necessidade de empresas estarem em conformidade com diferentes normas de qualidade, desde que isso seja feita de maneira rápida e de baixo custo. A oportunidade de investir tempo e pessoal apenas na fase de modelagem do processo da organização é atraente nesse sentido, embora o MDD seja uma abordagem que tem ainda muito a evoluir (PASTOR et al., 2008).

3

Trabalhos Relacionados

3.1 Introdução

Este capítulo trata dos trabalhos que tem relação com o escopo deste trabalho. Ou seja, foram consideradas ferramentas de software para avaliação de processos que são baseadas em uma ou mais modelos ou normas de qualidade. O processo de desenvolvimento das ferramentas também será descrito, bem como sua arquitetura, para posterior comparação com a solução proposta por este trabalho.

3.2 ImPProS

Com a evolução dos ambientes de desenvolvimento de software, estes passaram a suportar, além da função de desenvolvimento de software, também as funções associadas de gerência e garantia da qualidade durante o ciclo de vida do software.

No entanto, em alguns casos, percebe-se ao longo da execução do processo a partir destes ambientes de desenvolvimento que sua implementação nem sempre está adequada à realidade da organização ou do projeto desenvolvido por esta. Assim, para ajudar uma organização na implementação progressiva de um processo de software, é útil fornecer apoio automatizado por meio de um ambiente capaz de suportar as fases que a literatura especializada propõe como necessárias.

Essa implementação é considerada progressiva pelo fato de que as experiências adquiridas na definição, simulação, execução e avaliação do processo são incorporadas ao processo que está sendo implementado.

Nesse contexto, surge o ambiente ImPProS (OLIVEIRA; VASCONCELOS; ROUILLER, 2005) tem como objetivo principal apoiar a implementação de um processo de software em uma organização. Dentro deste contexto podem ser caracterizados como seus objetivos específicos:

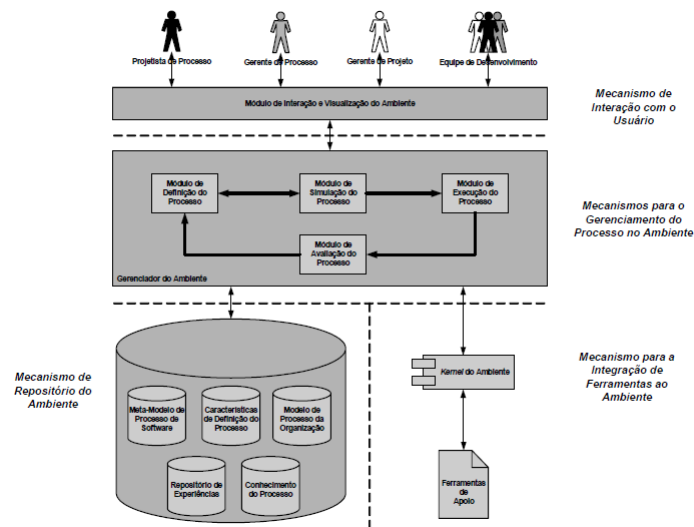
- Especificar um meta-modelo de processo de software a fim de definir uma terminologia única entre os vários modelos de qualidade de processo de software existentes, para uso do ambiente em seus serviços providos;

- Apoiar a definição de um processo de software para organização;
- Permitir a modelagem e instanciação deste processo;
- Permitir a simulação do processo a partir das características instanciadas para um projeto específico;
- Dar apoio à execução do processo de software tomando como base uma máquina de inferência;
- Possibilitar a avaliação dos critérios do processo de software;
- Apoiar a melhoria contínua do processo de software e o reuso através da realimentação e coleta das experiências aprendidas.

Vale ressaltar que todos os objetivos listados acima foram adaptados a partir da estrutura que compõe o meta-processo de software que flexibiliza e aumenta o nível de automação para execução de processos. Além do meta-modelo, a características propostas para a implementação de um processo de software e do ciclo de vida para melhoria contínua de processo definido pelo Modelo IDEAL (MCFEELEY, 1996) também foram considerados. Para alcançar estes objetivos o ambiente foi concebido para adotar a arquitetura apresentada na Figura 3.1.

O ImPProS é composto por 4 submecanismos, cuja integração está descrita na Figura 3.1:

- **Mecanismo de Interação com o Usuário:** neste mecanismo o foco está em prover aos usuários diferentes visões da mesma informação sendo definida e especificada, provendo interação para diferentes usuários do ambiente, ou seja, trabalha com as características da usabilidade no ambiente.
- **Mecanismo para o Gerenciamento do Processo no Ambiente:** este mecanismo possui a responsabilidade de prover os serviços (módulos de definição, simulação, execução e avaliação do processo de software) especificados ao ambiente de forma automatizada, ou seja, possibilitar que os usuários do ambiente executem suas funções tendo como referencial um guia.
- **Mecanismo de Repositório do Ambiente:** o foco deste mecanismo está em prover ao ambiente o sistema de gerenciamento dos seus objetos a partir de bases de dados que provejam o controle de evolução e manutenção dos componentes do processo de software.
- **Mecanismo para Integração de Ferramentas ao Ambiente:** este mecanismo provê a integração do ambiente com outras ferramentas de apoio ao processo de software e à execução do projeto de software, possibilitando a automação de atividades definidas no processo e a execução de alguns módulos do ambiente.

Figura 3.1: Arquitetura do ambiente ImPProS

Fonte: OLIVEIRA; VASCONCELOS; ROUILLER, 2005

Além da composição de mecanismos, o ImPProS possui um ambiente cooperativo, que inclui 9 ferramentas principais (OLIVEIRA; VASCONCELOS; ROUILLER, 2005):

- ProDefiner: fornece a definição de processo de software a partir da análise de características específicas e aprendizagem com outras definições;
- ProSimulator: torna possível a simulação de processo de software instanciado a partir de um plano de execução do processo, com o objetivo de prever problemas;
- ProEnacter: ele permite a execução automatizada e acompanhada de um processo de software pela equipe do projeto;
- ProEvaluator: fornece a avaliação da execução de processos de software a partir da análise de critérios qualitativos e quantitativos;
- ProImprove: torna possível a execução sistemática das atividades de melhoria do processo de software, a partir do modelo IDEAL;
- ProAnalyser: permite a análise e tomada de decisão sobre os itens de avaliação que compõem o processo de software;
- ProReuse: proporciona a reutilização da definição de projeto na execução do processo de software e sua adaptação ao contexto de uso;
- ProKnowledge: torna possível a coleta, análise e uso do conhecimento obtido durante a execução do processo de software;
- ProConverter: fornece a conversão de componentes do processo de software a partir de estruturas de normas e modelos de qualidade, e sua mapeamento.

A ferramenta ProEvaluator possui um propósito semelhante ao deste trabalho, e suas duas versões desenvolvidas serão analisadas nas duas seções seguintes.

3.3 ProEvaluator

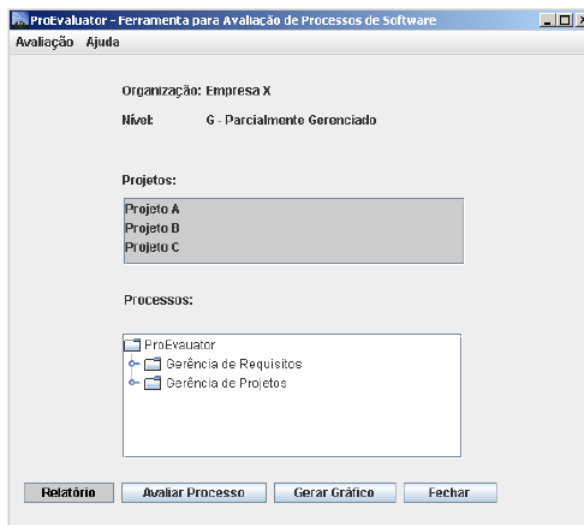
A ferramenta ProEvaluator (SOTERO, 2009) está dentro do ambiente ImPProS (OLIVEIRA; VASCONCELOS; ROUILLER, 2005), cujo objetivo é implantar a implementação progressiva de processo de software dentro de uma organização. O ProEvaluator é uma ferramenta para avaliação de processos de software que dá apoio à automação do Método de Avaliação do MPS.BR, o MA-MPS.

O objetivo da ferramenta ProEvaluator é automatizar algumas atividades relacionadas à avaliação de um processo de desenvolvimento de software. Ela oferece suporte às fases: “Preparar para a realização da avaliação”, “Realizar a Avaliação” e “Documentar os Resultados da Avaliação”, definidas no Processo de Avaliação do MPS.BR. Podemos listar os principais objetivos da ferramenta ProEvaluator, sendo estes:

- Oferecer suporte ao processo de avaliação interna das organizações. Ou seja, ela pode ser utilizada para a organização conceber um diagnóstico dos seus processos com o objetivo de melhorar a qualidade dos mesmos. A partir daí, ela saberá o que falta para atingir um determinado nível de maturidade. Nessa pré-avaliação a organização identifica os pontos fortes e pontos a melhorar do seu processo. Dependendo do resultado dessa avaliação a organização decidirá se vale a pena investir numa avaliação oficial ou não;
- Oferecer suporte à atividade de avaliação de processos de software definida no Método de Avaliação MPS.BR. A ferramenta poderá ser utilizada pelas Instituições Avaliadoras credenciadas pela SOFTEX para realizar a avaliação oficial MPS.BR.

Para criar uma avaliação no ProEvaluator é necessário fornecer o nome da Organização, os projetos que serão avaliados e o nível almejado pela Organização. A partir daí a avaliação já foi criada e pode ser reaberta através da ferramenta sempre que necessário. Por isso, o início do fluxo de atividades da ProEvaluator começa com uma decisão: Criar ou Abrir uma avaliação já existente. A partir daí é exibida uma tela contendo as informações da avaliação como os processos que serão avaliados, os projetos e o nível almejado, como podemos ver na Figura 3.2. O próximo passo é escolher o processo onde serão cadastradas as evidências do processo da Organização.

O processo é exibido com seus resultados esperados e seus atributos de processo. O avaliador deve apenas selecionar o resultado que deseja cadastrar a evidência, escolher o tipo de evidência (Direta, Indireta ou Afirmativa) e realizar o cadastro. Ele deve fazer isso para todos os resultados esperados de todos os processos e de seus atributos.

Figura 3.2: Ferramenta ProEvaluator - Tela Tela Geral de Avaliação

Fonte: SOTERO, 2009

Após cadastrar as evidências, o avaliador procede com a atribuição de conceitos para os resultados esperados dos processos e de seus atributos. Nessa hora ele pode descrever os pontos fortes e oportunidades de melhoria do processo da Organização. Essas informações estarão presentes no relatório final da avaliação. Depois que todos os resultados esperados de um processo forem avaliados, o sistema calcula automaticamente o conceito do mesmo. Esse conceito pode ser: Satisfeito, Não Satisfeito ou Fora de Escopo, conforme critérios definidos no MA-MPS. Podemos ver parte desse fluxo na Figura 3.3.

Figura 3.3: Ferramenta ProEvaluator - Tela de Avaliação de Processo

Fonte: SOTERO, 2009

Se o avaliador não concordar com a nota gerada automaticamente pela ferramenta, ele pode alterar a mesma. Depois que o avaliador estiver de acordo com o conceito do processo, ele deve gerar o gráfico dos resultados do processo e de seus atributos. Esse gráfico mostra o conceito de cada resultado esperado, como visto na Figura 3.4. Para finalizar a avaliação é

necessário criar o relatório final da avaliação. Tanto o usuário comum quanto o avaliador podem realizar essa operação.

Figura 3.4: Ferramenta ProEvaluator - Tela de Gráficos dos Resultados Esperados do Processo



Fonte: SOTERO, 2009

3.4 ProEvaluator 2.0

A ferramenta ProEvaluator foi originada do trabalho descrito na seção 3.2, a qual só contemplava a automação do método de avaliação do MPS.BR, o MA.MPS.

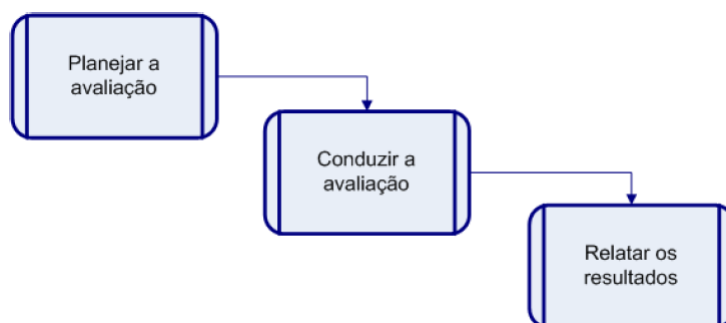
O ProEvaluator 2.0 (XAVIER, 2007) considera um modelo genérico de avaliação de processos aderente aos principais métodos de avaliação: SCAMPI, MA.MPS e ISO/IEC 15504.

A ferramenta ProEvaluator 2.0 fornece as funcionalidades necessárias para execução de um fluxo genérico de avaliação, baseado na norma ISO/IEC 15504, que consequentemente se adequa ao CMMI e ao MPS.BR. Para elaboração deste fluxo foram estudados os métodos de avaliação de processos, já apresentados no Capítulo 2, e foi possível verificar que os mesmos têm uma série de atividades em comum. O fluxo genérico que a ProEvaluator fornece suporte está representado na Figura 3.5:

O processo “Planejar a avaliação” consiste na geração de todos os artefatos de planejamento. Tem como principais atividades:

- Analisar o escopo da avaliação bem como seus requisitos;
- Estabelecer o time envolvido com a avaliação;
- Elaborar um plano de avaliação, contendo todo o detalhamento de como deve ser realizada a condução da auditoria e relato de seus resultados. Este plano de avaliação deve conter toda a estratégia que será utilizada para coleta das evidências;

Figura 3.5: Ferramenta ProEvaluator 2.0- Fluxo de Avaliação



Fonte: XAVIER, 2007

- Analisar as evidências enviadas previamente pelo patrocinador da empresa;

O processo “Conduzir avaliação” consiste na realização da mesma. O processo “Relatar os resultados” consiste na finalização da avaliação. Neste processo final será apresentado se a organização atende aos requisitos necessários do(s) modelo(s) ou norma(s) de referência que escolheu ser avaliada. As principais atividades desta fase são: Entregar os resultados da avaliação e Arquivar os resultados da avaliação.

Um dos objetivos já detalhados da ferramenta ProEvaluator 2.0 é apresentar um modelo genérico que suporte o relacionamento entre os principais modelos e normas de referência, de forma a facilitar o compartilhamento entre as evidências dos mesmos. Os modelos de referência utilizados foram CMMI, MPS.BR e ISO/IEC 15504.

Esta implementação multimodelo apresenta uma alternativa para a implementação de uma ferramenta minimamente aderente a vários modelos e normas de qualidade diferentes. Porém, a implementação pode estar comprometida se uma atualização for feita em pelo menos um dos modelos, exigindo tanto a revisão do modelo genérico quanto a recodificação da ferramenta.

Caso a organização esteja utilizando a ferramenta pela primeira vez é necessário cadastrar o(s) modelo(s) aos quais deseja-se avaliar a sua aderência. Para isto é necessário cadastrar o framework que está associado ao modelo inicialmente. Isso pode ser visto na Figura 3.6.

Após cadastrar o framework que o modelo pertence, é necessário o cadastro do modelo. O cadastro do modelo pode ser visualizado na Figura 3.7

O próximo passo do planejamento é o cadastro das informações referentes à Avaliação. Estas informações consistem no cadastro do método de avaliação que será utilizado, dos dados da avaliação em si, das atividades e dos avaliadores. Após estes cadastros é possível iniciar a avaliação da organização.

A partir desta definição, a ProEvaluator 2.0 fornece o cadastro de avaliação da prática, onde é possível cadastrar evidências, observações referentes à avaliação da prática e suas deficiências. Os itens produtos de trabalho e subpráticas são informativos.

Caso a prática avaliada tenha sido mapeada como equivalente a algum item de outro modelo, as evidências do item do outro modelo irão aparecer como forma de sugestão de

Figura 3.6: Ferramenta ProEvaluator - Tela de Cadastro de Framework

Home Planejamento Execução Relatórios Advanced Options

Home Frameworks Framework Novo Framework

Dados do Framework

Id:

Nome:

Descrição:

1. Cadastrar Framework

1

Home Planejamento Execução Relatórios Advanced Options

Home Editor de Modelos Novo Modelo

Editor de Modelos

Modelos Frameworks

Id	Nome	Descrição
1	CMMI Framework	The CMMI Framework is the basic structure that organizes CMMI components and combines them into CMMI constellations and models

Fonte: XAVIER, 2007

Figura 3.7: Ferramenta ProEvaluator - Tela de Cadastro de Modelo

ProEvaluator

Home Planejamento Execução Relatórios Advanced Options

Home Editor de Modelos Novo Modelo

Dados do modelo

Id:

Framework Name:

Acronym:

Name:

Description:

Editor de Modelos

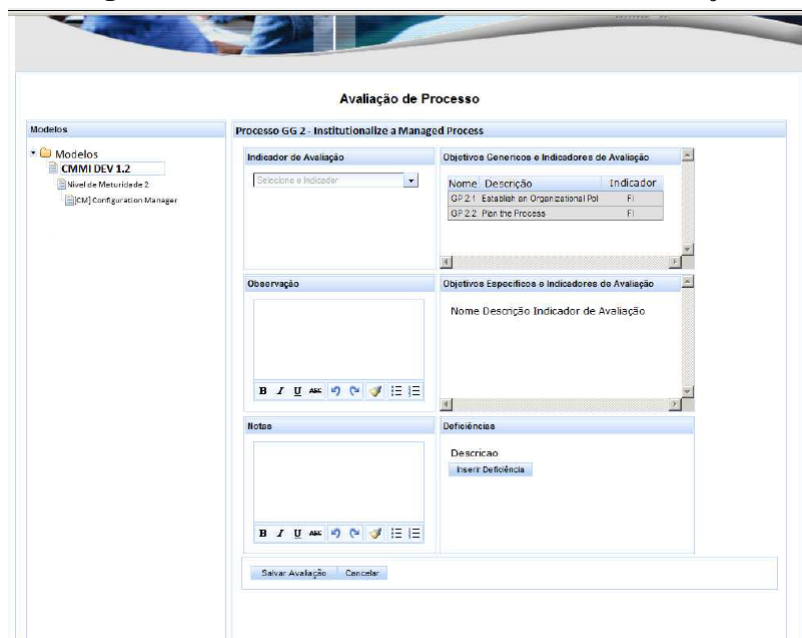
Modelos Frameworks

Id	Framework	Acronym	Model Name
1	CMMI Framework	CMMI DEV V1.2	CMMI for Development, Version 1.2

Fonte: XAVIER, 2007

evidências para este item. Ou seja, itens semelhantes de diferentes modelos poderão compartilhar evidências a partir da ferramenta. O cadastro da avaliação da prática poderá ser visualizado na Figura 3.8

Figura 3.8: Ferramenta ProEvaluator - Tela de Avaliação



Fonte: XAVIER, 2007

3.5 FAPS

A ferramenta FAPS (Ferramenta de Avaliação de Processos de Software) (THIRY et al., 2008) foi desenvolvida pela equipe do LQPS (Laboratório de Qualidade de Software) da UNIVALI (Universidade do Vale do Itajaí) em cooperação com a II-MPS.BR Incremental Tecnologia Ltda para apoiar a avaliação integrada de processos de software. Foi desenvolvida considerando o método de avaliação MA-MPS, os requisitos estabelecidos no CMMI-ARC e a parte 2 da ISO/IEC 15504. Desta forma, a avaliação integrada suporta os modelos MPS.BR, CMMI e norma ISO/IEC 15504, respectivamente.

O modelo CMMI e a norma ISO/IEC 15504 foram considerados pelo seu reconhecimento internacional. O modelo MPS.BR foi também considerado porque além de ter sido desenvolvido de forma compatível ao modelo e norma supracitados é nacionalmente reconhecido e está também alinhado à realidade das MPEs (Micro e Pequenas Empresas) de software brasileiras.

Com o uso da FAPS, é possível obter, de modo semi-automático, a conformidade de uma avaliação realizada com base na norma ou modelo escolhido em relação a outro modelo de qualidade, dentre os três modelos que compõem o escopo do FAPS. Também, é possível ter apoio no gerenciamento dos documentos utilizados durante uma avaliação e auxílio para avaliações recorrentes em uma mesma unidade organizacional. Para isto, a ferramenta FAPS

permite a reutilização de evidências coletadas e de resultados da avaliação de um dos modelos e norma suportados pela ferramenta.

Após uma avaliação, a FAPS permite que um controle histórico da sequência de avaliações de uma mesma unidade organizacional seja armazenado. Isto auxilia a manter baselines do processo, que posteriormente podem vir a ser comparadas estabelecendo um parâmetro de quanto de melhoria os processos de software adquiriram com o trabalho de implementação.

A FAPS permite que somente usuários cadastrados façam login no sistema e que apenas os avaliadores de uma determinada avaliação tenham acesso aos dados da avaliação conduzida por eles. Portanto, a FAPS gerencia o acesso à sua base de dados permitindo que sejam definidos perfis e privilégios. A FAPS disponibiliza um cadastro para empresas, unidades organizacionais, projetos, equipes de avaliadores e ainda as avaliações em si.

Após efetuar os cadastros mencionados anteriormente (preparação inicial), a unidade organizacional pode iniciar o cadastramento das evidências. Na Figura 3.9, pode-se visualizar 4 evidências cadastradas para o resultado GPR17 do processo Gerência de Projetos do MPS.BR. Neste exemplo, está sendo considerado o nível G de maturidade.

Figura 3.9: FAPS - Tela de Evidências

Evidência	Tipo	Fonte	Org	P1	P2
Plano de Projeto (Seção 4.1 - Recursos Humanos)	Direta	GP			
Papéis e responsabilidades gerais	Direta	GP			
Banco de currículos dos colaboradores	Direta	FGC			
Registro do apontamento de horas	Indireta	FRH			

Fonte: THIRY et al., 2008

Os processos com seus respectivos resultados são apresentados na parte esquerda da tela, assim como os atributos de processo também com seus resultados. O usuário pode selecionar qualquer resultado para visualizar as evidências a ele relacionadas. Na parte superior da tela, é apresentado o resultado selecionado com seu detalhamento.

A estrutura central é similar àquela encontrada nas planilhas de avaliação do MA-MPS e do SCAMPI. Além da coluna para descrever a evidência, existem colunas para indicar o tipo de evidência (direta, indireta ou afirmação) e a fonte da evidência. Também pode-se verificar a existência da coluna organizacional e uma coluna para cada projeto cadastrado (o número

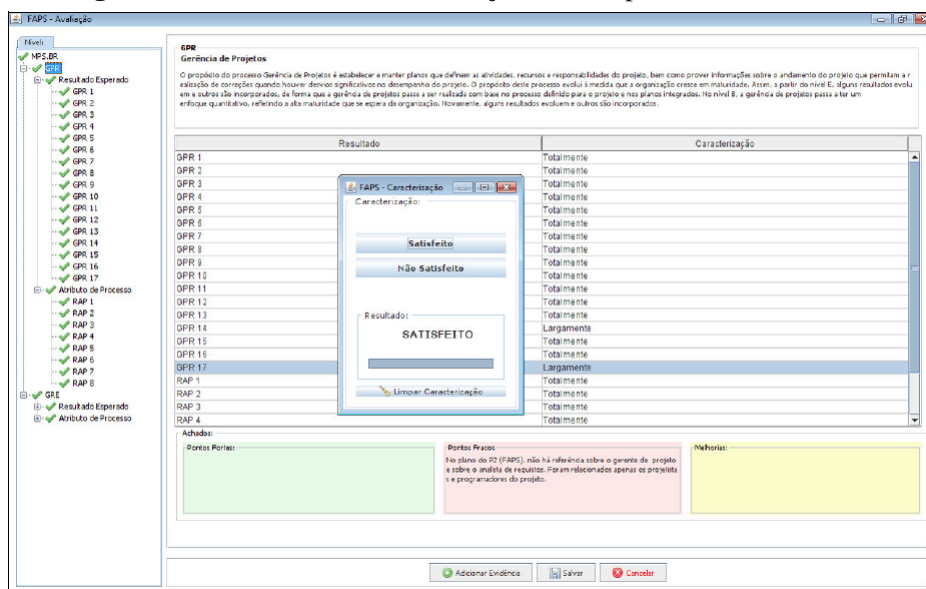
de projetos pode ser variável). As evidências cadastradas são também armazenadas em um repositório (lateral direita da tela), permitindo sua reutilização em outros resultados.

Durante a avaliação apoiada pela FAPS, é possível alterar a cor das colunas organizacional e projetos. A cor branca significa que a evidência ainda não foi avaliada, a cor verde indica que a evidência foi julgada adequada, a cor amarela indica alguma fraqueza na evidência e a cor vermelha é usada para indicar que a evidência não está adequada.

Na parte inferior da tela, é possível visualizar duas áreas distintas, sendo também possível armazenar os achados (pontos fortes, fracos e oportunidades de melhoria). Além destes relacionados com os resultados, a FAPS permite ainda cadastrar achados sobre o modelo/norma ou sobre aspectos gerais da organização. Há legendas com a indicação das cores, com os nomes dos projetos associados a P1, P2, etc. No cadastro de um projeto, é possível indicar se ele está concluído ou não. Esta informação é mostrada aqui ao lado da sigla do projeto. A legenda também inclui a descrição das siglas utilizadas nas fontes das evidências.

Para auxiliar a edição das evidências, é possível também selecionar múltiplas evidências para exclusão. Dando sequência ao processo de avaliação, a equipe de avaliadores pode registrar também na FAPS a caracterização de cada resultado e processo sendo avaliado. Este registro é apresentado a seguir na Figura 3.10 que mostra a tela de caracterização de processo.

Figura 3.10: FAPS - Tela de Avaliação de Componentes do Processo



Fonte: THIRY et al., 2008

3.6 Requisitos Coletados

A análise de trabalhos realizados em um contexto de avaliação de processos de software foi utilizada para levantamento de requisitos relacionados à solução como produto de software.

Definições como plataforma, linguagem de programação e até padrão de interface foram definidos após a análise destes trabalhos.

A forma com que as ferramentas utilizam modelos foi analisada de forma prática, ou seja, em termos de como a ferramenta utiliza os artefatos de entrada. Além disso, foram coletados os requisitos que contemplam como os dados das avaliações são gerenciados e exibidos ao usuário.

3.7 Considerações Finais

As ferramentas descritas neste capítulo tem como função servir como parâmetro tanto para a construção da solução proposta por este trabalho, quanto para a avaliação final do mesmo. Elementos como interface, processo de desenvolvimento e como os modelos ou normas de qualidade foram considerados no desenvolvimento dos trabalhos citados foram considerados na elaboração e desenvolvimento da solução proposta por este trabalho.

É importante destacar que alguns dos trabalhos não representam necessariamente ferramentas de avaliação, porém os mesmos foram considerados por conterem elementos relacionados à modelagem ou à utilização de modelos para avaliação de processos de software, como é o caso do Imppros.

4

Processo de Desenvolvimento

4.1 Concepção

Para iniciar o processo de desenvolvimento, temos a fase de concepção. Esta considerou, principalmente, os objetivos listados no Capítulo 1, quanto os requisitos elicitados após o estudo da teoria na literatura (Capítulo 2) e ferramentas já desenvolvidas com o mesmo propósito (Capítulo 3).

4.1.1 Objetivos

Muitos dos requisitos da ferramenta proposta foram baseados nos objetivos da mesma. Foram os seguintes, descritos no Capítulo 1:

- Ser gerada a partir de modelos: Utilizar os conceitos de MDD para gerar as ferramentas de avaliação;
- Genérica para gerar diferentes ferramentas baseadas em diferentes modelos: Dado um modelo e seu método de avaliação (CMMI + SCAMPI, ou MPS-BR e MA-MPS), a ferramenta deve ser gerada baseada no modelo de entrada.
- Suportar tanto a norma quanto o método de avaliação: A geração da ferramenta deve se basear nos dois modelos dados como entrada;
- Ser simples de executar, deixando a complexidade para a modelagem; Extrair o máximo possível de informações dos artefatos de entrada. Utilização do EMOF para geração de modelos;
- Utilizar modelos MOF gerados pelo Eclipse Modeling Framework: O EMF foi escolhido com os objetivos de padronizar os modelos que vão ser recebidos como entrada, bem como ser um padrão para documentação dos mesmos.
- Requerer o mínimo possível de passos manuais para gerar a ferramenta: Uso intensivo de templates e bibliotecas visando agilizar a geração da ferramenta;

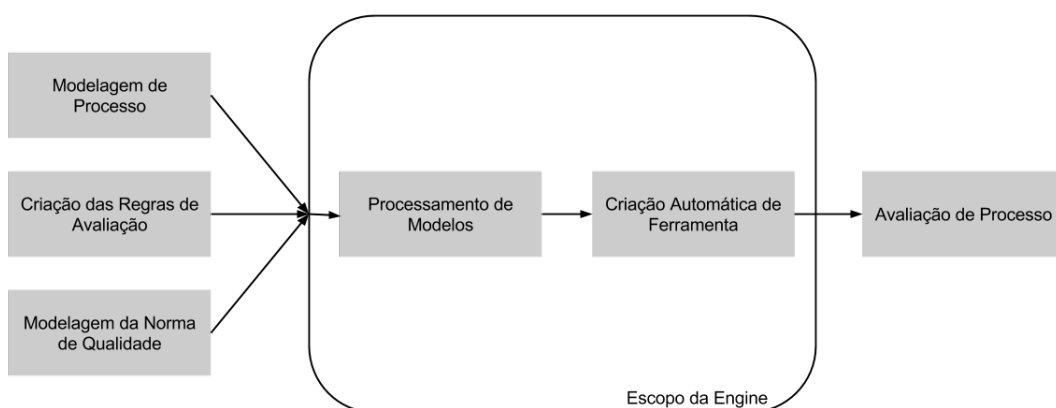
- Deve gerar uma ferramenta simples, de fácil entendimento e curva de aprendizagem baixa: Algo que possa ser preenchido com TABs, de maneira a agilizar o processo de preenchimento;

De acordo com o primeiro objetivo, o processo de desenvolvimento da solução proposta por este trabalho utiliza a abordagem MDD para a geração de ferramentas de avaliação. Assim, dois conceitos fundamentais foram considerados para a criação da engine: os modelos a serem utilizados como entrada no processo, e a geração de uma ferramenta automaticamente, baseada nos modelos.

Quanto ao segundo e terceiro objetivos, é preciso deixar claro que a solução proposta não contém em seu escopo a atividade de modelar a norma de qualidade ou método de avaliação. Ou seja, a engine se baseia em artefatos prontos, previamente modelados. A engine também possui um conjunto de regras sobre a modelagem, que serão detalhadas posteriormente. Estas precisam ser seguidas para que a engine reconheça de maneira correta os dados inseridos.

Uma descrição gráfica em alto nível das atividades que compõem o escopo da ferramenta pode ser vista na Figura 4.1

Figura 4.1: Atividades realizadas pela Engine



A utilização de modelos e do MDD em si se deu por conta do alto poder de criação de ferramentas novas, dado o seu baixo custo de desenvolvimento de uma solução (PASTOR et al., 2008). Além disso, a constante revisão e atualização dos artefatos de entrada prolongam a vida útil dos mesmos, evitando longos e custosos processos de modelagem. Essas atividades de trabalhar constantemente com os artefatos de entrada se dá pela forma como estes são fundamentais para o processo de desenvolvimento, precisando estar sempre aptos a gerar a versão mais recente da ferramenta em que estes artefatos se baseiam.

O quarto objetivo leva a outro ponto importante, considerado no desenvolvimento da solução, que é a escolha das linguagens utilizadas para modelar os artefatos de entrada. A utilização de XMI (OMG, 2014a) na descrição dos modelos se deu pela ampla utilização desta linguagem como saída de ferramentas de modelagem que exportam seus modelos, como as

ferramentas desenvolvidas utilizando o Eclipse Modeling Framework - EMF (FOUNDATION, 2010).

Para a descrição dos métodos de avaliação, foi utilizada a linguagem OCL (Object Constraint Language) (OMG, 2014b), que cria restrições que um determinado conjunto de elementos deve obedecer. Podemos considerar um método de avaliação como um conjunto de restrições a qual um processo deve obedecer para que sua avaliação seja positiva.

Os dois últimos objetivos são contemplados utilizando-se boas práticas de programação para web, como o padrão MVC. A escolha da solução online se deve ao fato de esta tecnologia ser multi-plataforma, não demandar instalação e estar disponível em qualquer lugar, sendo acessível via browser, ainda que dependa inteiramente de conexão com a internet para funcionar.

Dada a escolha da tecnologia web, a linguagem escolhida foi o PHP (Hypertext Preprocessor) (GROUP, 2001), que conta com uma ampla comunidade e está a mais de 19 anos no mercado. As funções nativas da linguagem, bem como as bibliotecas disponíveis serviram de base para a implementação da abordagem MDD utilizada.

Devido ao fato de que uma avaliação em geral dura dias ou até semanas (CHEN; HOI; XIAO, 2011), foi preciso criar um mecanismo de persistência para as ferramentas geradas pela engine. Apesar de já ser menos custoso criar e manter uma ferramenta automaticamente através de modelos, também há um baixo custo associado a utilização de uma ferramenta já criada e que já contenha os dados da avaliação em curso.

4.1.2 Requisitos

A partir dos objetivos listados na seção 4.1 e da análise das ferramentas descritas no Capítulo 3, uma lista de requisitos foi definida.

Outro ponto importante a se considerar na Elicitação dos Requisitos da solução foi que apenas a atividade de execução da avaliação foi considerada para este fim. Toda a parte de planejamento (agendamento de entrevistas, cronograma, realização de entrevistas, coleta de dados, etc) não faz parte do escopo da solução. Isso se deu pelo fato de que o escopo da ferramenta gerada precisou ser restringido, devido aos modelos utilizados como entradas. Com mais atividades associadas, os artefatos de entrada precisariam conter mais informações, ou até mesmo demandar outros artefatos.

A fase dos relatórios finais e encerramento da avaliação também não é contida pelo escopo, já que se limita ao armazenamento e compartilhamento dos dados, não estando adequada aos padrões de divulgação de resultados.

É importante destacar que existem três tipos de requisitos considerados para o desenvolvimento da solução proposta por este trabalho: os Requisitos Gerais, os Requisitos Específicos para Engine e os Requisitos Específicos para as Ferramentas (Tabela 4.3).

4.1.2.1 Requisitos Gerais

Os Requisitos Gerais representam o conjunto de requisitos que precisam ser satisfeitos por qualquer área ou módulo da solução proposta. O desenvolvimento de todo e qualquer módulo foi direcionado para a satisfação dos Requisitos Gerais, assim como escolha de bibliotecas e criação de templates. Os Requisitos Gerais estão descritos na Tabela 4.1.

Tabela 4.1: Lista dos Requisitos Gerais

Requisito	ID	Prioridade	Descrição
Ferramenta Web	RG01	Alta	A solução precisa estar disponível para acesso via internet a qualquer momento
Interface Amigável	RG02	Alta	A interface com o usuário deve ser amigável, de modo que seu uso seja eficiente
Fácil Aprendizado	RG03	Alta	Todas as ações da solução precisam ser intuitivas e de escopo simples
Persistência	RG04	Média	É necessário que as ferramentas e dados gerados pelos usuários sejam armazenados para uso posterior.

4.1.2.2 Requisitos Específicos da Engine

Os Requisitos Específicos da Engine representam o conjunto de requisitos que são exclusivos da engine de geração de ferramentas. Em geral, são os requisitos relacionados à abordagem MDD utilizada, aos artefatos de entrada e mecanismos de identificação de metamodelos e processos utilizados para a geração das ferramentas. Os Requisitos Específicos da Engine estão descritos na Tabela 4.2.

É importante destacar a prioridade "Média" atribuída ao Requisito Específico da Engine RE05. Esse requisito recebeu essa classificação devido à recomendação de que a atividade de modelagem seja realizada com bastante cuidado antes da utilização da engine. Com isso, foi definido como requisito que a verificação seja feita apenas para a validação dos modelos entre si, não sintaticamente de acordo com as regras do EMF.

4.1.2.3 Requisitos Específicos das Ferramentas

Os Requisitos Específicos da Ferramenta representam o conjunto de requisitos que são exclusivos das ferramentas que serão geradas pela engine. Em geral, são os requisitos relacionados à interface, à persistência e exibição dos dados informados durante a avaliação. Os Requisitos Específicos da Ferramenta estão descritos na Tabela 4.3.

Tabela 4.2: Lista dos Requisitos Específicos da Engine

Requisito	ID	Prioridade	Descrição
Leitura de Modelos XMI	RE01	Alta	O modelo recebido como entrada deve estar no formato .xmi
Leitura de Regras OCL	RE02	Alta	O conjunto de regras recebido como entrada deve estar no formato .ocl
Geração automática de Classes	RE03	Alta	As classes das ferramentas serão geradas exclusivamente nos artefatos de entrada, sem interação com o usuário
Uso de templates	RE04	Média	Utilização de elementos pré-construídos, a fim de agilizar e padronizar o desenvolvimento automático
Verificação da corretude dos artefatos	RE05	Média	Os artefatos são verificados entre si para avaliar sua corretude.
Suporte a vários modelos e normas de qualidade	RE06	Alta	O usuário deverá definir que modelo ou norma será contemplado pela ferramenta apenas pela escolha dos artefatos de entrada
Identificação única das ferramentas geradas	RE07	Alta	Uma ferramenta deve ser identificada unicamente por um ID fornecido pela própria engine
Repositório de ferramentas geradas	RE08	Alta	As ferramentas geradas devem ser armazenadas persistentemente, possibilitando acesso posterior
Suporte de atualização a uma ferramenta específica	RE09	Média	Através do ID requerido pelo RE07, o usuário pode gerar uma nova versão de uma ferramenta previamente criada.

4.2 Elaboração

Após a fase de concepção da solução proposta, foi iniciada a elaboração da solução. Nessa fase, foram definidos os workflows das duas partes da solução, bem como foi definida sua arquitetura. Após essas definições, foi iniciado o desenvolvimento propriamente dito da solução.

4.2.1 Workflows

Após a elicitação dos requisitos, estes foram organizados de forma a criar o workflow geral da solução que este documento descreve. Como existem dois produtos gerados, a Engine de Geração de Ferramentas e as ferramentas geradas em si, dois workflows foram criados para descrever a sequência de atividades.

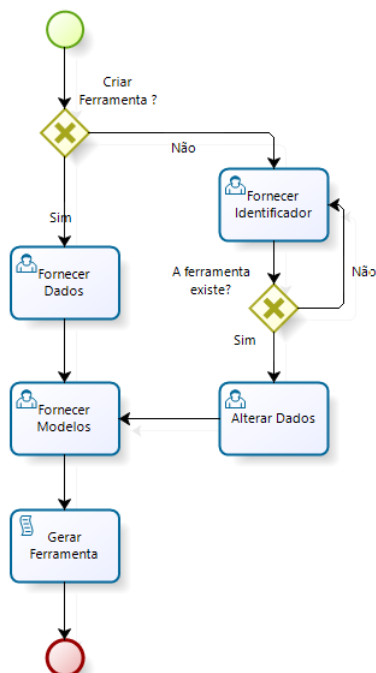
Tabela 4.3: Lista dos Requisitos Específicos da Ferramenta

Requisito	ID	Prioridade	Descrição
Suporte à atribuição dos valores ou conceitos	RF01	Alta	Os valores atribuídos aos componentes do processo serão inseridos na ferramenta
Suporte à edição dos valores ou conceitos	RF02	Alta	Os valores atribuídos aos componentes do processo poderão ser editados na ferramenta
Suporte a adição de evidências	RF03	Média	Evidências da avaliação poderão ser cadastradas na ferramenta
Suporte a edição de evidências	RF04	Média	Evidências da avaliação poderão ser editadas na ferramenta
Persistência dos dados da avaliação	RF05	Alta	Os dados da avaliação em curso poderão ser acessados a qualquer momento, mesmo que a ferramenta seja acessada novamente
Computação automática dos resultados	RF06	Alta	Uma vez finalizada a entrada dos dados, a ferramenta deve automaticamente aplicar as regras do método e gerar o resultado
Identificação dos dados da avaliação	RF07	Alta	Dados referentes à avaliação como um todo podem ser adicionados à ferramenta
Geração de Relatórios	RF08	Alta	Relatórios parciais ou totais podem ser gerados a qualquer momento pela ferramenta
Exportação de Relatórios	RF09	Média	Os relatórios gerados podem ser exportados em formato PDF.

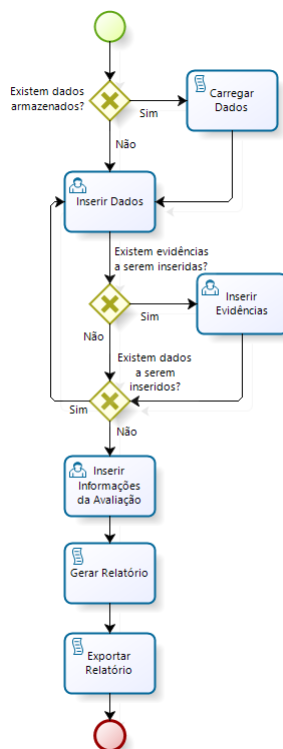
4.2.1.1 Workflow da Engine

O fluxo de funcionamento da Engine (como visto na Figura 4.2), de maneira geral, é bastante simples. Inicialmente, o usuário tem duas opções de fluxo: o de criar uma nova ferramenta ou o de atualizar uma já existente. Caso escolha o segundo, será requerido a identificação única da ferramenta gerada (atividade Fornecer Identificador), para fins de busca. Tendo encontrado a ferramenta, a engine irá permitir a alteração dos dados da ferramenta (atividade Alterar Dados), para então receber os modelos fornecidos pelo usuário (atividade Fornecer Modelos). Daí então automaticamente a engine gera a ferramenta desejada (atividade automática Gerar Ferramenta).

Caso o fluxo escolhido tenha sido o de criar nova ferramenta, o fluxo é bastante semelhante ao de atualização, exceto que os dados serão fornecidos pela primeira vez (atividade Fornecer Dados), para a partir daí os modelos serem lidos e a ferramenta gerada (atividade Fornecer Modelos em diante).

Figura 4.2: Workflow da Engine de Geração de Ferramentas

4.2.1.2 Workflow das Ferramentas

Figura 4.3: Workflow da Ferramenta Gerada

O fluxo da ferramenta apresenta-se um pouco mais complexo do que o da engine. Inicialmente, a ferramenta automaticamente verifica a existência de dados já cadastrados por um

avaliador. Caso existam dados, estes são carregados na interface do sistema (atividade Carregar Dados), e a interface já abastecida desses dados será exibida permitindo a inserção de outros dados (atividade Inserir Dados). Se não existirem dados, uma interface sem dados de avaliação será exibida para dar início à inserção dos dados (atividade Inserir Dados).

Durante a inserção dos dados, evidências podem ser associadas a esses dados inseridos (atividade Inserir Evidências). Após a inserção de dados e evidências, o usuário pode inserir os dados referentes à atividade global de avaliação (atividade Inserir Informações da Avaliação), para a geração dos resultados e dos relatórios (atividade de Gerar Relatório). Há também a possibilidade do usuário exportar o relatório (atividade Exportar Relatório), para armazenamento offline dos dados e resultados.

4.2.2 Visão Geral e Arquitetura

A organização da arquitetura da engine se deu pelo agrupamento de classes em módulos, tendo como seus componentes classes que possuíssem forte coesão. Além disso, os módulos foram organizados para seguirem a sequência do fluxo mostrado na Figura 4.1.

Outra característica da organização da Engine foi o uso de templates para geração de classes. Utilizando conceitos de orientação a objetos, as classes são pré-escritas através destes templates, para que, independente dos artefatos fornecidos como entrada, as classes tenham atributos e métodos semelhantes entre si.

A arquitetura da solução pode ser detalhada dividindo-a em três partes: os Artefatos de Entrada, a Engine em si e as Ferramentas de Avaliação geradas. Uma arquitetura detalhada, incluindo o relacionamento entre as partes e seus módulos pode ser vista na Figura 4.4

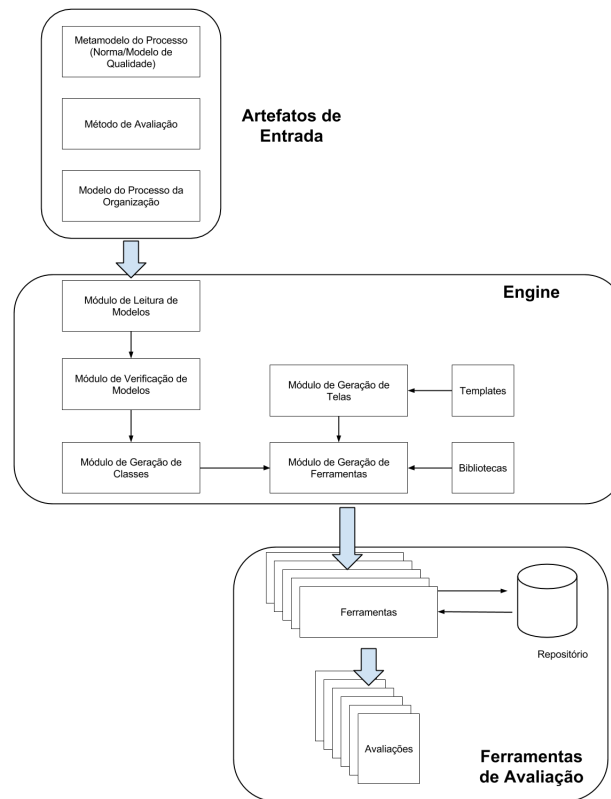
4.2.3 Artefatos de Entrada

Três artefatos compõem as entradas para a engine: O metamodelo da norma ou modelo de qualidade, o modelo do processo (que consiste em uma instância do metamodelo) e do método de avaliação, descrito por meio de regras OCL.

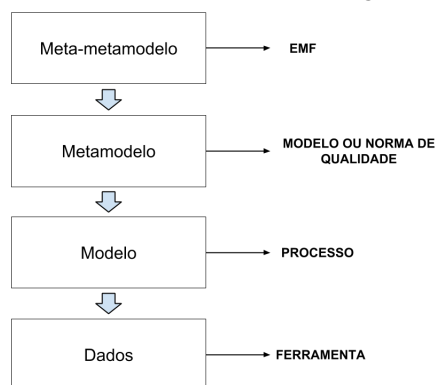
Como estamos utilizando a abordagem MDD, é importante contextualizar a utilização de metamodelos, tanto do modelo ou norma a ser utilizado para avaliação, quanto do EMF, do qual os metamodelos são instanciados.

A OMG definiu um padrão para MDE (Model-Driven Engineering), chamado MOF (Meta Object Facility) (OMG, 2014c). Esta define uma arquitetura com 4 camadas de modelos.

Esta arquitetura de modelos tem como uma de suas vantagens a facilidade para substituição de modelos na arquitetura, pela alteração de modelos de níveis superiores (ATKINSON; KÜHNE, 2003). Esta facilidade é muito interessante visto que um dos objetivos da engine, de poder gerar ferramentas de acordo com vários modelos, seja algo que é inerente à abordagem MDD.

Figura 4.4: Módulos e Artefatos que compõem todo o ambiente da Engine

Podemos fazer um paralelo entre a arquitetura do MOF e a estrutura dos artefatos da engine, onde cada artefato representa um nível, como visto na Figura 4.5

Figura 4.5: Relacionamento entre os modelos da Engine de acordo com o MOF

4.2.3.1 Metamodelo do Processo

Através do metamodelo, o usuário informa à engine como os componentes do modelo ou norma de qualidade ao qual as regras de avaliação pertencem se relacionam. Relacionamentos entre elementos, tipos e coleções são definidas neste artefato. Consequentemente, como as

classes que serão geradas pela engine irão se relacionar entre si, de acordo com as regras de orientação à objetos.

Podemos considerar o metamodelo como o artefato mais importante do conjunto. Como foi definido em um dos objetivos específicos deste trabalho, a complexidade da geração de ferramentas foi transferida para a modelagem inicial. Ora, se o metamodelo define como os objetos vão se relacionar dentro da ferramenta gerada, qualquer mudança que seja feita no metamodelo pode ter impacto negativo na mecânica global de avaliação da ferramenta. Além disso, os dois outros artefatos utilizados como entradas estão diretamente relacionados com a estrutura do metamodelo, e alterações neste podem invalidar os outros artefatos e comprometer o resultado da execução da engine.

Para descrever o metamodelo, utilizamos a linguagem XMI (OMG, 2014a), que é utilizado em geral pelas ferramentas desenvolvidas através do EMF como formato de saída (FOUNDATION, 2010).

4.2.3.2 Modelo do Processo

Outro artefato a ser utilizado pela engine é o modelo do processo a ser avaliado. Este relaciona como conjuntos de elementos definidos no metamodelo serão utilizado na prática pelo processo a ser avaliado. É importante frisar a relação direta entre os dois modelos, onde o modelo do processo é uma instância do metamodelo. Com isso, quando há uma atualização no modelo ou norma, um novo modelo de processo é gerado, inerentemente.

É este modelo que vai ser submetido às regras do método de avaliação, servindo como base da criação do conjunto de objetos que a ferramenta gerada pela engine vai gerenciar e associar os conceitos atingidos pelos processos durante a avaliação.

O modelo do processo também faz parte da arquitetura MOF, estando relacionado diretamente com o modelo ou norma de qualidade a ser contemplado pela ferramenta.

Considerando este cenário de mudanças de metamodelos e/ou de modelos de processo, a engine foi concebida também com a possibilidade de um histórico de versões ser criado para cada par metamodelo-modelo. Ou seja, de acordo com algum identificador de versão de metamodelo, o avaliador é capaz de navegar entre as várias versões de ferramentas já geradas para aquela organização. Isto pode auxiliar o avaliador em como o processo da empresa evoluiu ao longo das versões do metamodelo.

Pelo mesmo motivo que o XMI foi utilizado para descrever o metamodelo, também foi escolhido para a definição do modelo do processo.

4.2.3.3 Método de Avaliação

Um dos componentes do método de avaliação é um conjunto de regras que regem a graduação dos componentes do processo de acordo com a norma ou modelo de qualidade. Por isso, foi preciso transformar esse componente do método em um artefato que possa ser lido pela

engine e transformado na mecânica que vai relacionar os dados da avaliação na qual a ferramenta a ser criada vai ser utilizada.

De novo, é preciso observar que as regras precisam fazer referência a elementos contidos no metamodelo do processo, para que toda a mecânica seja criada de forma coerente com a norma utilizada. Além disso, as classes criadas pela engine para representar elementos da avaliação também se baseiam em determinadas regras fornecidas pelo arquivo de regras OCL.

Na Tabela 4.4, é mostrada um resumo do propósito do artefato na engine, bem como o seu formato esperado:

Tabela 4.4: Resumo dos artefatos de entrada da Engine

Artefato	Formato	Descrição
Metamodelo do Processo	XMI	Descreve os componentes do processo de acordo com a norma ou modelo de qualidade
Modelo do Processo	XMI	Descreve como os componentes descritos no metamodelo interagem entre si dentro do contexto da organização
Método de Avaliação	OCL	Descreve as regras para graduação da implementação dos componentes do metamodelos no modelo do processo

4.2.4 Engine

A engine em si, que compõe o *core* da solução, é composta de 5 módulos e de dois conjuntos separados: um de templates e outro de bibliotecas. As bibliotecas foram utilizadas principalmente para criação da interface global da aplicação (engine e ferramentas), bem como a ferramenta de preenchimento de templates com dados fornecidos como entrada. Os módulos da engine vão ser descritos nas seções seguintes.

4.2.4.1 Módulo de Leitura de Modelos

Este módulo é responsável pela leitura dos artefatos de entrada da engine. Aqui é feita a verificação dos tipos dos modelos, se estão de acordo com o metamodelo e entre si, bem como se o artefato foi inserido no campo correto da engine. Foi criado em um módulo separado da verificação para facilitar a inserção e remoção de artefatos de entrada.

4.2.4.2 Módulo de Verificação de Modelos

Neste módulo, após a leitura dos modelos, os mesmos são verificados de acordo com o artefato imediatamente superior na arquitetura do MOF, dos níveis 1 ao 3. Neste módulo, o

modelo do processo é confrontado com seu metamodelo (ambos recebidos como entrada), e o metamodelo do processo é confrontado com o metamodelo do EMF. Além disso, o conjunto de regras fornecido também é validado com os modelos.

4.2.4.3 Módulo de Geração de Classes

Após a validação dos modelos entre si e de acordo com o metamodelo do EMF, as informações referentes às classes que irão compor a ferramenta são fornecidas aos templates, que irão gerar as ferramentas. Neste módulo é criada a parte lógica da ferramenta, onde os valores fornecidos pelo avaliador serão processados para compor o resultado final da atividade.

Como foi mencionado anteriormente, os templates utilizados por esse módulo respeitam os princípios básicos de orientação à objeto, como encapsulamento, baixo acoplamento, coesão alta.

Outro fator importante considerado para a criação desse módulo é que, ao gerar a ferramenta, esta respeite o padrão MVC para aplicações web. E este requisito é atendido pela engine, pois a lógica controladora da aplicação é criada independentemente da parte de visão.

4.2.4.4 Módulo de Geração de Telas

Também criadas com a ajuda de templates, as telas das ferramentas geradas são incorporadas às ferramentas partindo deste módulo. Este módulo não é gerado automaticamente de acordo com os modelos, como o módulo anterior. Porém, este foi criado para facilitar a manutenção da parte de visão da engine como um todo, assim como de ferramentas a serem geradas e as já existentes.

É importante destacar também que a usabilidade global da solução proposta por este trabalho concentra-se neste módulo. Uso de bibliotecas de interface, estilo de telas, mensagens e qualquer outra interação com o usuário pode ser alterada com a manutenção deste módulo.

4.2.4.5 Módulo de Geração de Ferramentas

Este módulo realiza a montagem da ferramenta a ser apresentada ao usuário. Ela faz a conexão da lógica criada no Módulo de Criação de Classes com as telas criadas no módulo correspondente.

Além disso ela cria a estrutura que identifica unicamente as ferramentas geradas, pelo par metamodelo-organização. Ainda num contexto de uma avaliação por uma organização, temos a criação da estrutura de persistência dos dados da avaliação corrente, dado o tempo de realização das mesmas (que podem durar dias, ou até semanas). As ferramentas geradas serão explicadas mais detalhadamente nas seções seguintes.

4.2.5 Ferramentas de Avaliação

As ferramentas de avaliação geradas pela engine apresentam todas a mesma estrutura, atuando como uma interface dos dados armazenados da avaliação com o avaliador, além de contar com a lógica de geração dos resultados da avaliação.

Cada ferramenta gerada tem seu controlador gerado, para se adequar totalmente ao modelo fornecido como entrada. As suas telas, porém, são criadas a partir de templates padrões.

Apesar de estarem contidas na mesma arquitetura que a engine e serem geradas por ela, as ferramentas podem ser consideradas como uma aplicação à parte. Seu funcionamento, a partir do momento que é gerada, é completamente independente da engine, embora estejam ligados por questões de plataforma.

4.3 Avaliação

Após a elaboração da solução proposta, uma avaliação da mesma se fez necessária. Para realizar essa avaliação, foram considerados os requisitos elicitados na fase de elaboração. Neste primeiro momento, um exemplo de uso baseado em um caso real foi utilizado. No Capítulo 5 é realizada uma avaliação, realizada por usuários que tem experiência na atividade de avaliar processos de software.

4.3.1 Exemplo de Uso

4.3.1.1 Entradas Utilizadas

As entradas utilizadas para demonstração são baseadas no modelo CMMI, utilizando o método de avaliação SCAMPI. No SCAMPI, a avaliação é realizada a partir das práticas até os objetivos. Assim, para caracterizar o nível de implementação de uma prática (SP ou GP), as seguintes categorias são usadas:

- **Totalmente Implantado (FI)** - Os requisitos de avaliação estão presentes e são considerados adequados para demonstrar a implementação prática, além de não terem sido encontrados pontos fracos;
- **Largamente Implementado (LI)** - Os requisitos de avaliação estão presentes e considerado adequado para demonstrar a implementação prática. No entanto, um ou mais pontos fracos são encontrados;
- **Parcialmente implementado (PI)** - Alguns ou todos os dados necessários estão ausentes ou considerados inadequados, alguns dados fornecidos (se existirem) sugerem que aspectos da prática são implementados, e um ou mais pontos fracos são encontrados; ou os dados fornecidos à equipe de avaliação possuem conflitos, ou seja,

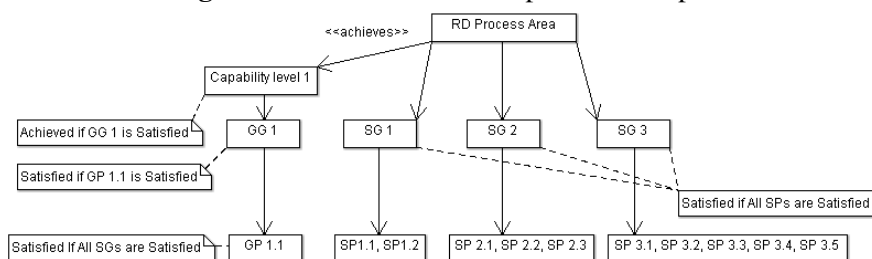
alguns dados indicam que a prática é implementada e outros dados indicam que a prática não é implementada, e também, uma ou mais deficiências são observadas;

- **Não implementado (NI)** - Alguns ou todos os dados necessários estão ausentes ou considerados inadequados, os dados fornecidos (se existirem) não apoiam a conclusão de que a prática é implementada, e um ou mais pontos fracos são encontrados.

Com base nas notas definidas para a prática, cada objetivo (SG ou GG) é classificado como: 1) Satisfeito - se e somente se todas as práticas associadas são classificadas como LI ou FI, e a agregação das fraquezas associadas com o objetivo não ter um impacto negativo significativo sobre a realização do objetivo, ou; 2) Insatisfeito - se pelo menos uma das práticas associadas tem um grau diferente de LI ou FI. Infelizmente este método de classificação não é completamente objetivo, porque depende da experiência do avaliador.

Finalmente, com base nas notas definidas para os objetivos, o nível de capacidade de um PA é definida. Por exemplo, para cumprir com o nível de capacidade 1, um processo deve satisfazer o objectivo genérico (GG), associada ao nível de , que tem apenas uma prática genérica (GP 1.1) que solicita todos os SGs associados sejam também satisfeitos, para que a PA seja atendida (se, pelo menos, um dos dos SGs não é satisfeito, o PA é considerado como tendo nível de capacidade 0). Para a avaliação dos níveis de capacidade maior do que 1, um PA deve satisfazer o GG associado ao nível específico, o que impõe outras exigências, e os GGs associados aos níveis mais baixos. A Figura 4.6 mostra o nível de capacidade 1 da área de processo Desenvolvimento de Requisitos (RD).

Figura 4.6: Modelo utilizado para os exemplos



Fonte: O Autor

4.3.1.2 Exemplo 1

Como explicado anteriormente, este exemplo visa apresentar a estrutura da engine, de acordo com o que foi descrito com o Capítulo 4. Serão descritas e analisadas a usabilidade, disposição dos elementos e a mecânica da interface com o usuário.

Iniciaremos o exemplo com a tela inicial da engine na Figura 4.7. Ao acessar a URL da ferramenta, o usuário vai ter visão dessa interface, que se divide em 3 partes distintas, sendo duas delas identificadas por seus respectivos títulos: Informações sobre a ferramenta e Informações sobre a norma.

Figura 4.7: Tela inicial da Engine

Fonte: O Autor

Como a primeira das três, temos um menu na parte superior da tela, que representa as opções de ações que o usuário pode realizar na engine, como detalhado na Figura 4.8. As opções são:

- **Criar:** Nesse menu, que representa a tela inicial do sistema, temos os passos necessários para gerar uma nova ferramenta;
- **Atualizar:** Nesse menu, temos os passos necessários para gerar de uma nova ferramenta a partir de uma ferramenta previamente gerada;
- **Contato:** Menu destinado à informações e ações de feedback sobre o uso da engine;

Figura 4.8: Menu da Engine

Fonte: O Autor

A seguir, o usuário insere informações referentes ao contexto da avaliação, como a versão do metamodelo utilizado e o identificador da ferramenta. Ao inserir um Identificador, a engine

automaticamente gera um outro identificador (campo Permalink), que será utilizado como URL de acesso para a ferramenta gerada. A Figura 4.9 mostra um exemplo da inserção dos dados.

O campo "Permalink" não permite edição, pois apenas um conjunto reduzido de caracteres pode ser utilizado em sua composição. Se o campo fosse deixado para edição livre, a URL que seria associada a ele poderia conter caracteres que comprometeriam o acesso a ferramenta.

Além disso, os dois outros campos ("Identificador" e "Versão do Metamodelo") são obrigatórios, ou seja, a ferramenta não será gerada se estes campos não estiverem preenchidos.

Figura 4.9: Dados da ferramenta ser gerada

Fonte: O Autor

Na parte inferior da tela, os campos correspondentes a cada um dos artefatos de entrada estão posicionados. São campos para arquivos, e, ao submeter qualquer arquivo, a engine automaticamente verifica a sua extensão e retorna para o usuário se o arquivo é ou não do tipo esperado. Na Figura 4.10 vemos o exemplo onde cada arquivo foi posicionado em seu respectivo local, com as extensões corretas.

Figura 4.10: Dados da norma de qualidade e do processo a ser avaliado pela ferramenta

Fonte: O Autor

Por fim, ao clicar em gerar, o usuário tem a geração da ferramenta sendo automaticamente realizada. O resultado está na Figura 4.11.

4.3.1.3 Exemplo 2

Depois de gerada a ferramenta, a interface da engine direciona o usuário diretamente para aquela, com o objetivo de apresentar o resultado. Como visto na Figura 4.11, a ferramenta

Figura 4.11: Ferramenta gerada pela Engine

Avaliação - Teste1

Informações sobre a avaliação

Avaliador:

Data de Início:

Data de Término:

Projeto: +

Projeto	Excluir
Projeto 1	
Projeto 2	

Process Area 1 (Process Area responsible for bla bla bla) 3

Capability Level 1 (Description here) 2

Fonte: O Autor

organiza as informações da avaliação num cabeçalho, e depois exibe os outros elementos abaixo deste.

O usuário precisa então preencher estes campos para identificar a avaliação e pode submeter os dados para geração do relatório. Os dados são o nome do avaliador, a data de início e término e um conjunto de um ou mais projetos que serão contemplos pela avaliação. Podemos ver um conjunto de dados de exemplo na Figura 4.12.

Figura 4.12: Dados referentes a uma avaliação inseridos

☒ORG:
☒Bcard:
☒PRONAF:

GPR 2. As tarefas e os produtos de trabalho do projeto sao dimensionados utilizando metodos apropriados.

As evidencias apresentadas para este resultado permitem assegurar que o tamanho e/ou a complexidade das tarefas e dos artefatos gerados no projeto foram estimados utilizando metodos adequados (ex: baseados na EAP ou estrutura equivalente, em tecnicas de estimativa ou em dados historicos)?

Resultado: Cadastrar Evidências

☒ORG:
☒Bcard:
☒PRONAF:

GPR 3. O modelo e as fases do ciclo de vida do projeto sao definidos.

As evidencias apresentadas para este resultado permitem assegurar que o modelo do ciclo de vida do projeto foi definido, indicando suas fases, as relacoes de sequencia e interdependencia entre elas?

Resultado: Cadastrar Evidências

☒ORG:
☒Bcard:
☒PRONAF:

GPR 4. (Ate o nivel F) O esforco e o custo para a execucao das tarefas e dos produtos de trabalho sao estimados com base em dados historicos ou referencias tecnicas.

Fonte: O Autor

A tabela de projetos simplesmente lista os projetos que serão avaliados dentro da organização. Para tal, o usuário fornece o nome do projeto e clica no botão à direita do campo. O valor então é acrescido na tabela, podendo ainda ser removido posteriormente. O campo não permite a inserção de strings vazias.

Seguindo adiante, temos a avaliação dos componentes do projeto. Essa parte foi projetada

para que a visualização das áreas possam ser omitidas, para que elementos que não estejam em foco no momento possam ser ocultos para uso posterior.

A Figura 4.13 apresenta a estrutura de campos para a avaliação. O avaliador tem como opções as graduações definidas no arquivo OCL do método de avaliação, através de uma lista *dropdown*. O campo não foi deixado como texto livre (como os campos da área de Informações sobre a avaliação) para que o padrão definido pela engine fosse seguido durante todo o fluxo da avaliação.

Figura 4.13: Interface para graduação de elementos do processo

The screenshot displays a web interface for evaluating process elements. It features three sections, each corresponding to a GPR (Goal, Purpose, Result) item. Each section includes a title, a descriptive text, a 'Resultado' dropdown menu, and a 'Cadastrar Evidências' button. The sections are:

- GPR 2:** As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados. The text below states: 'As evidências apresentadas para este resultado permitem assegurar que o tamanho e/ou a complexidade das tarefas e dos artefatos gerados no projeto foram estimados utilizando métodos adequados (ex: baseados na EAP ou estrutura equivalente, em técnicas de estimativa ou em dados históricos)?'.
- GPR 3:** O modelo e as fases do ciclo de vida do projeto são definidos. The text below states: 'As evidências apresentadas para este resultado permitem assegurar que o modelo do ciclo de vida do projeto foi definido, indicando suas fases, as relações de sequência e interdependência entre elas?'.
- GPR 4:** (Ate o nível F) O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas.

Each section also includes a set of checkboxes for 'ORG', 'Bcard', and 'PRONAF', each followed by a 'Select...' dropdown menu.

Fonte: O Autor

É importante lembrar que os dados são persistidos à medida que são digitados e que periodicamente os dados são salvos pela própria ferramenta. Na prática, isto representa a possibilidade de avaliações serem realizadas durante um intervalo longo de tempo. Esta funcionalidade, combinada com o ambiente web, torna a ferramenta poderosa no sentido de ser acessível de vários locais (incluindo mobile, ainda que o projeto não seja diretamente voltado para esta plataforma).

Ainda em relação à Figura 4.13, além de graduar os elementos do processo, a ferramenta permite que o avaliador cadastre evidências para reforçar essa avaliação. Ao lado da lista de avaliações, temos o botão Cadastrar Evidência.

Ao clicar no botão, a ferramenta exibe a tela de evidências associadas a cada elemento. Se já houver alguma evidência cadastrada, a mesma é exibida em uma tabela (semelhante à de projetos) com as evidências e suas fontes.

O cadastro é feito com campos de texto, tendo os mesmos a restrição de não aceitar strings vazias. O usuário, ao finalizar, clica no botão de Adicionar Evidência para adicionar a mesma a tabela. A janela de cadastro de evidências pode ser vista na Figura 4.14.

Após avaliar todos os componentes do processo e cadastrar todas as evidências, se cabível, o avaliador submete a avaliação à ferramenta, como visto na Figura 4.15. Esta, então, sumariza os dados e os organiza em forma de relatório. Os dados, então, estão armazenados na

Figura 4.14: Tela de cadastro de evidências

Evidência	Fonte	Excluir
ED - Evidência 1	Fonte 1	
EI - Evidência 2	Fonte 2	

Fonte: O Autor

web, podendo ser consultados a qualquer momento através da interface da ferramenta. Ou seja, a ferramenta funciona como um relatório permanente de avaliações, que poderão ser editados ou revisados constantemente, sem a preocupação de gerenciar numerosos arquivos para cada avaliação.

Figura 4.15: Botão para enviar a avaliação

Fonte: O Autor

Após o envio, o usuário é direcionado para a tela que sumariza os resultados fornecidos. Além de exibir todos os dados, como visto na Figura 4.16, o usuário tem a opção de retornar à avaliação (botão Nova Avaliação) ou gerar um relatório da avaliação em formato de arquivo PDF (botão Gerar Relatório).

O relatório a ser gerado tem a sua estrutura semelhante à tela de resultados. A funcionalidade foi adicionada para permitir ao usuário armazenar localmente os resultados, caso seja necessário. Além disso, é uma funcionalidade presente na maioria das ferramentas de avaliação

existentes.

Figura 4.16: Sumarização dos dados da ferramenta

3. Resultados

Resultados:

▲ Informações sobre a avaliação

Avaliador: Luiz Liborio

Data de Início: 03/07/2014

Data de Término: 06/07/2014

Projeto(s): Projeto 1, Projeto 2

▼ Process Area 1 (Process Area responsible for bla bla bla) 3

▼ Process Area 2 (Description here) 3

Nova Avaliação

Gerar Relatório

4.4 Considerações Finais

Apesar de ter sido pensada apenas como uma engine de geração de ferramentas, a solução se dividiu em duas partes integradas, porém distintas: a engine em si, e a estrutura da ferramenta a ser gerada. As mecânicas de persistência e atualização dessas ferramentas foram pensadas para que a ferramenta se mostrasse útil além do contexto de uma avaliação simples, podendo também ser eficiente para avaliações mais complexas e que levem mais tempo para serem concluídas. No capítulo seguinte serão apresentados dois exemplos de uso da engine e da ferramenta.

A ferramenta apresentada tem como objetivo ser um meio útil e eficiente de criar relatórios de avaliação de processos. Como foi dito anteriormente, a combinação de persistência de dados com o ambiente web torna a ferramenta de avaliação um importante meio de armazenamento remoto de dados, que podem ser consultados a qualquer momento por um avaliador.

5

Avaliação da Pesquisa

5.1 Introdução

As ferramentas geradas pela engine têm como um dos seus principais objetivos de serem úteis também para quem já utiliza os meios existentes de avaliação. Portanto, os requisitos combinados de uma ferramenta web e de meios de avaliação podem ser testados com resultados reais de avaliação. Então, de acordo com o comparativo dos resultados obtidos pela ferramenta gerada com os dados de uma avaliação real, podemos ter uma medida de quanto a engine que gera ferramentas genéricas está próximo da realidade.

Vimos no Capítulo 4 um exemplo de utilização da engine com um subconjunto de elementos do CMMI, e o método de avaliação SCAMPI. Para este capítulo, utilizaremos uma avaliação baseada no MPS.BR com o MA-MPS. Os dados utilizados como entrada serão detalhados na seção seguinte.

Além disso, a solução proposta por este trabalho será comparada aos trabalhos relacionados que foram discutidos no Capítulo 3. Esta comparação foi feita baseando-se nos requisitos propostos para a engine e para a ferramenta gerada.

Finalmente, uma pesquisa realizada com usuários avaliadores de processo irá mostrar como a engine e a ferramenta são avaliadas do ponto de vista prático, por profissionais experientes que lidam com variadas formas de avaliação no cotidiano.

5.2 Comparativo com Avaliação Real

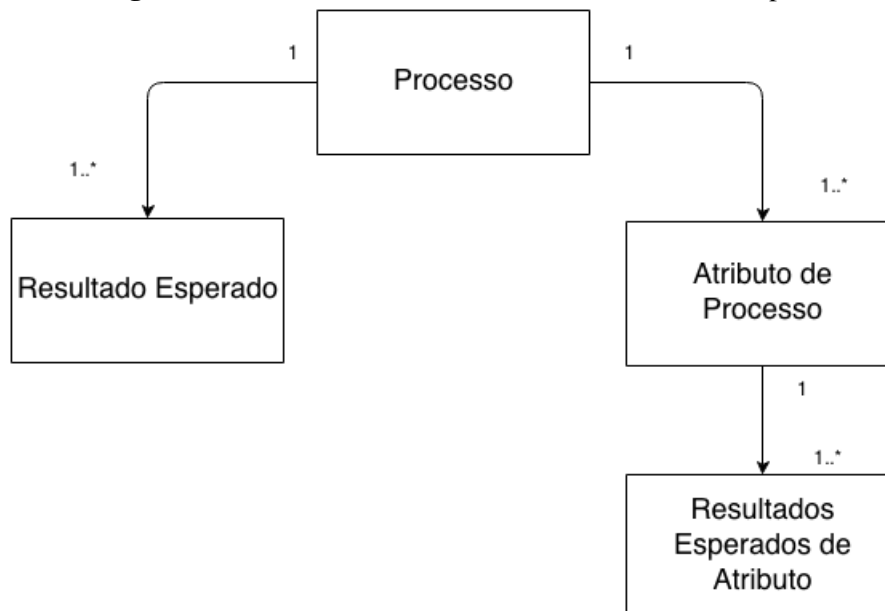
5.2.1 Entrada

A avaliação utilizada para comparação foi realizada em uma organização real. Por razões óbvias de confidencialidade, os dados da organização foram omitidos, restando apenas os dados que realmente interessam para a avaliação.

A avaliação realizada foi para o nível de maturidade G do MPS.BR. Os processos e atributos de processo desse nível são Gerência de Requisitos e Gerência de Projetos, de acordo

com o Guia do MPS.BR. Na Figura 5.1, temos o metamodelo gerado a partir da planilha de avaliação fornecida como exemplo:

Figura 5.1: Metamodelo dos Dados de Entrada do Exemplo



Fonte: O Autor

5.2.1.1 Processos

- **Gerência de Projetos (GPR):** O propósito do processo Gerência de Projetos é estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto. O propósito deste processo evolui à medida que a organização cresce em maturidade. Assim, a partir do nível E, alguns resultados evoluem e outros são incorporados, de forma que a gerência de projetos passe a ser realizada com base no processo definido para o projeto e nos planos integrados. No nível B, a gerência de projetos passa a ter um enfoque quantitativo, refletindo a alta maturidade que se espera da organização. Novamente, alguns resultados evoluem e outros são incorporados.
- **Gerência de Requisitos (GRE):** O propósito do processo Gerência de Requisitos é gerenciar os requisitos do produto e dos componentes do produto do projeto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto.

Cada um dos processos possui um conjunto de Atributos de Processo e de Resultados, de acordo com cada nível de maturidade.

5.2.1.2 Atributos de Processo

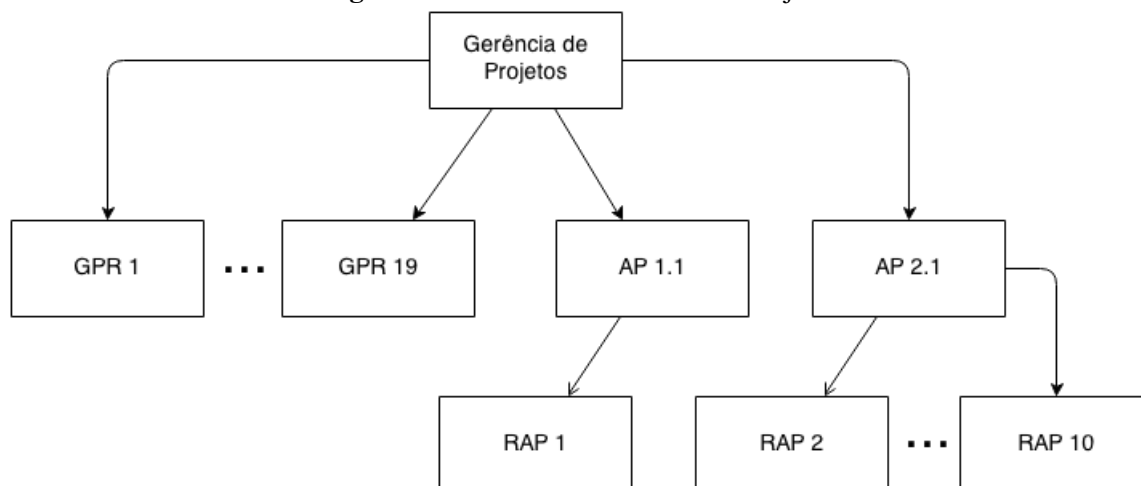
- **AP 1.1** : Este atributo evidencia o quanto o processo atinge o seu propósito.
- **AP 2.1**: Este atributo evidencia o quanto a execução do processo é gerenciada.

5.2.2 Modelo dos Processos Avaliados

Após a criação do metamodelo de entrada, foi preciso modelar os dados da planilha em entidades que receberão os dados da avaliação a ser realizada. Portanto, cada processo do nível G de maturidade do MPS foi instanciado como uma entidade do modelo de entrada da ferramenta. As representações gráficas podem ser vistas em Figura 5.2 e Figura 5.3.

O processo de Gerência de Projetos possui 19 Resultados associados, além dos dois Atributos de Processo associados ao Nível G de maturidade do MPS.BR. O Atributo de Processo 1.1 possui o Resultado RAP 1 associado, e o Atributo de Processo 2.1 possui os RAPs de 2 a 10.

Figura 5.2: Processo - Gerência de Projetos



Fonte: O Autor

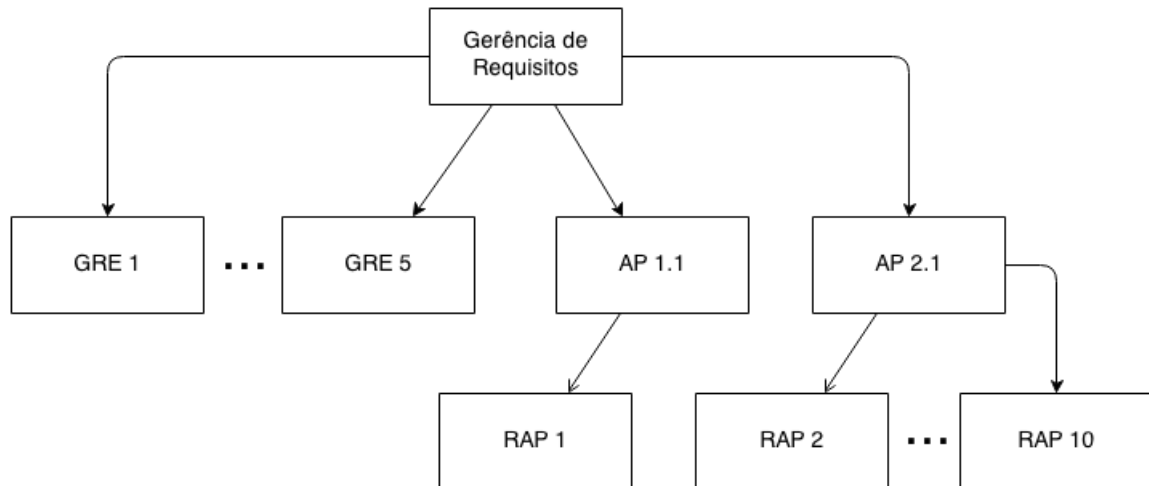
O processo de Gerência de Requisitos, por sua vez, possui 5 Resultados associados, além dos dois Atributos de Processo associados ao Nível G de maturidade do MPS.BR, sendo semelhantes, portanto ao de Gerência de Projetos.

5.3 Planilha de Avaliação

Na Figura 5.4 temos uma parte da chamada Planilha de Indicadores, utilizada na avaliação. Temos no exemplo a visualização do processo de Gerência de Projetos. Detalhes como Projetos (cada um contido em uma aba separada), Evidências e a Classificação de cada uma.

A planilha possui cada componente com a sua descrição que consta no Guia do MPS.BR. Ainda conta com uma aba à parte explicando como preencher a mesma. Ou seja, ainda que a

Figura 5.3: Processo - Gerência de Requisitos



Fonte: O Autor

Figura 5.4: Planilha de Indicadores utilizada como exemplo comparativo

Gerência de Projetos						
PREENCHIDO PELA EMPRESA						
Resultado esperado / evidências	Fonte da evidência	ORG	Brand	PRONAF	Projeto	Final
O propósito do processo Gerência de Projetos é estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto. O propósito deste processo evolui à medida que a organização cresce em maturidade. Assim, a partir do nível E, alguns resultados evoluem e outros são incorporados, de forma que a gerência de projetos passe a ser realizada com base no processo definido para o projeto e nos planos integrados. No nível B, a gerência de projetos passa a ter um enfoque quantitativo, refletindo a alta maturidade que se espera da organização. Novamente, alguns resultados evoluem e outros são incorporados.						
GPR 1. O escopo do trabalho para o projeto é definido.						
As evidências apresentadas para este resultado permitem assegurar que o escopo do projeto foi definido?						
ED - Visão de Projeto (VPJ)	Gerente de Projeto		X	X		
ED - Lista Inicial de Requisitos (LIR)	Analista de Requisitos		X	X		
EI - Declaração de Trabalho do Projeto (DT) - BCARD	Cliente		X			
EI - Solicitação de Serviço (SDS) - PRONAF	Cliente			X		
EI - Cronograma (CRD)	Gerente de Projeto		X	X		
Afirmação	(T,L,P,N,NA)		T	T		T
GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados. As evidências apresentadas para este resultado permitem assegurar que o tamanho e/ou a complexidade das tarefas e dos artefatos gerados no projeto foram estimados utilizando métodos adequados (ex: baseados na EAP ou estrutura						
ED - Planilha de Análise de Ponto por Função (FPA)	Gerente de Projeto		X	X		
ED - Política Organizacional (POE) - Fator de Ajuste = 1	Diretor		X			
EI - Cronograma (CRD)	Gerente de Projeto		X	X		
EI - Planilha de Esforço (PLE)	Gerente de Projeto		X	X		

Fonte: O Autor

planilha seja completamente editável, a forma de avaliação limita a criação de colunas e de itens a serem avaliados.

A planilha funciona graduando-se cada evidência de acordo com o(s) projeto(s) em que ela foi coletada. A partir dessa graduação de cada evidência, a graduação final do Resultado é uma composição do conjunto de evidências.

Figura 5.5: Planilha de Indicadores - Atributos de Processo

8	Afirmção	(T,L,P,N,NA)		T	T				T
9	AP 2.1 O processo é gerenciado								
	RAP 2. Existe uma política organizacional estabelecida e mantida para o processo.								
	As evidências apresentadas para este resultado permitem assegurar: (i) que foi definida uma política organizacional estabelecendo as expectativas da organização para a execução do processo e que esta política é conhecida e de fácil acesso aos interessados? (ii) que a política organizacional foi atualizada, quando necessário? (iii) que a política organizacional tem respaldo da alta administração (por exemplo, por meio de aprovação da alta administração)?								
1	ED - Política Organizacional Empresarial (POE)	Diretor	X						
2	EI - Email de divulgação	Gerente de Projeto	X						
3									
4									
5									
6	Afirmção	(T,L,P,N,NA)	T						T
7	RAP 3. A execução do processo é planejada.								
	As evidências apresentadas para este resultado permitem assegurar que existe um plano para a execução do processo?								
8	ED - Cronograma (CRD)	Gerente de Projeto	X	X					
9	EI - Plano de Projeto (PLP)	Gerente de Projeto	X	X					
1									
2									
3	Afirmção	(T,L,P,N,NA)	T	T					T
	RAP 4. (Para o nível G) A execução do processo é monitorada e ajustes são realizados								

Fonte: O Autor

Na Figura 5.5, temos uma amostra de como os Atributos de Processo são exibidos na planilha. A graduação dos Atributos é uma composição de graduações dos Resultados. Estes, por sua vez, são compostas por graduações de evidências coletadas no conjunto de projetos da organização.

5.3.1 Ferramenta Gerada

A ferramenta foi apresentada no capítulo 5, mas agora foi gerada com base em uma organização real, submetida a avaliação. Na Figura 5.6, temos a interface da ferramenta gerada para a inserção dos projetos que compõem o escopo da avaliação, tendo os projetos da planilha sido inseridos.

Figura 5.6: Ferramenta Gerada - Projetos

Avaliação - MPS.BR

Informações sobre a avaliação

Avaliador: Lutz Liborio

Data de Início: 03/07/2014

Data de Término: 09/07/2014

Projeto:

Projeto Excluir

ORG

Bcard

PRONAF

Prosseguir

Fonte: O Autor

Já na Figura 5.7, é exibida uma parte da tela, que corresponde à inserção de dados dos Resultados de Processo e de Atributos de Processo. Como foi mencionado anteriormente, a possibilidade de cadastrar evidências é habilitada para cada Resultado a ser inserido.

Figura 5.7: Ferramenta Gerada - Resultados e Atributos de Processo

Avaliação - MPS.BR

↑ Informações sobre a avaliação

Avaliador: Luiz Liborio
 Data de Início: 03/07/2014
 Data de Término: 09/07/2014
 Projetos: ORG, Bcard, PRONAF

↑ Gerencia de Projetos 21
 O propósito do processo Gerencia de Projetos é estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto. O propósito deste processo evolui à medida que a organização cresce em maturidade. Assim, a partir do nível E, alguns resultados evoluem e outros são incorporados, de forma que a gerencia de projetos passe a ser realizada com base no processo definido para o projeto e nos planos integrados. No nível B, a gerencia de projetos passa a ter um enfoque quantitativo, refletindo a alta maturidade que se espera da organização. Novamente, alguns resultados evoluem e outros são incorporados.

↑ GPR 1. O escopo do trabalho para o projeto é definido.
 As evidências apresentadas para este resultado permitem assegurar que o escopo do projeto foi definido?

Resultado:

☒ ORG:
☒ Bcard:

Fonte: O Autor

Cada um dos Resultados tem uma graduação, que se baseia na graduação dos projetos associados (no mínimo um). Para todo Resultado, há um botão para cadastro de evidências que suportem a avaliação de cada projeto. A Figura 5.8, a tela do relacionamento das evidências com os projetos e os resultados, tanto de atributos quanto de de processo.

Figura 5.8: Ferramenta Gerada - Tela de Cadastro de Evidências

☒ ORG:
☒ Bcard:
☒ PRONAF:

↑ GPR 2. As tarefas e os produtos são definidos.
 As evidências apresentadas para este resultado permitem assegurar que foram realizadas estimativas de custo e esforço para tarefas e produtos de trabalho com base em dados históricos ou métodos de estimativas e que foram documentadas as suas justificativas?

Resultado:
☒ ORG:
☒ Bcard:
☒ PRONAF:

↑ GPR 3. O modelo e as fases são definidos.
 As evidências apresentadas para este resultado permitem assegurar que foram realizadas estimativas de custo e esforço para tarefas e produtos de trabalho com base em dados históricos ou métodos de estimativas e que foram documentadas as suas justificativas?

Resultado:
☒ ORG:
☒ Bcard:
☒ PRONAF:

↑ GPR 4. (Ate o nível F) O esforço é definido.
 As evidências apresentadas para este resultado permitem assegurar que foram realizadas estimativas de custo e esforço para tarefas e produtos de trabalho com base em dados históricos ou métodos de estimativas e que foram documentadas as suas justificativas?

Resultado:
☒ ORG:
☒ Bcard:
☒ PRONAF:

Evidência:
 Fonte:
 Projetos:
 Selecionados: Bcard, PRONAF

Evidência	Fonte	Projetos	Excluir
EI - ED - Visão de Projeto (VPJ)	Gerente de Projeto	ORG, Bcard	X

Fonte: O Autor

Ao submeter a avaliação, a ferramenta exibe a tela de sumarização dos resultados e geração de relatórios. As evidências, com suas fontes, e a graduação dos Resultados são exibidas, bem como as descrições que foram definidas no Guia do MPS.BR. Esta tela pode ser vista na Figura 5.9

Figura 5.9: Ferramenta Gerada - Tela de Sumarização dos Resultados

Data de Inicio: 03/07/2014

Data de Término: 09/07/2014

Projeto(s): ORG, Bcard, PRONAF

Gerencia de Projetos

O proposito do processo Gerencia de Projetos e estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informacoes sobre o andamento do projeto que permitam a realizacao de correcoes quando houver desvios significativos no desempenho do projeto. O proposito deste processo evolui a medida que a organizacao cresce em maturidade. Assim, a partir do nivel E, alguns resultados evoluem e outros sao incorporados, de forma que a gerencia de projetos passe a ser realizada com base no processo definido para o projeto e nos planos integrados. No nivel B, a gerencia de projetos passa a ter um enfoque quantitativo, refletindo a alta maturidade que se espera da organizacao. Novamente, alguns resultados evoluem e outros sao incorporados.

GPR 1. O escopo do trabalho para o projeto e definido.

As evidencias apresentadas para este resultado permitem assegurar que o escopo do projeto foi definido? **Totalmente Implementado**

Evidência	Fonte
ED - Visão de Projeto (VPJ)	Gerente de Projeto
ED - Lista Inicial de Requisitos (LIR)	Analista de Requisitos
EI - Declaração de Trabalho do Projeto (DT) - BCARD	Cliente
EI - Solicitação de Serviço (SDS) - PRONAF	Cliente
EI - Cronograma (CRD)	Gerente de Projeto

GPR 2. As tarefas e os produtos de trabalho do projeto sao dimensionados utilizando metodos apropriados.

As evidencias apresentadas para este resultado permitem assegurar que o tamanho e/ou a complexidade das tarefas e dos artefatos gerados no projeto foram estimados utilizando metodos

Fonte: O Autor

É importante lembrar que os dados fornecidos para a ferramenta, caso seja necessário, podem acessados posteriormente pela mesma interface. Através do mecanismo de persistência, os mesmos podem inclusive ser alterados e resubmetidos.

5.3.2 Comparação da Planilha de Avaliação com a Ferramenta Gerada

Por serem duas formas estruturalmente diferentes de avaliação (planilha de dados e ferramenta web), apenas as áreas relacionadas a usabilidade e avaliação foram consideradas. A divisão dessa comparação nas áreas determinadas de acordo com os critérios definidos está na Tabela 5.1.

5.4 Comparativo com Outras Ferramentas

Outra forma de avaliar a ferramenta foi por meio de comparações com os trabalhos descritos no Capítulo 3. Estas outras ferramentas servem de comparativo com a solução proposta, de acordo com os objetivos descritos no Capítulo 1, devido ao escopo de execução de avaliação que as mesmas possuem. O comparativo está demonstrado na Tabela 5.2.

5.5 Avaliação por Usuários

5.5.1 Metodologia e Perfil dos Entrevistados

Uma avaliação da solução proposta foi realizada com 12 profissionais que estiveram presentes em alguma avaliação de processo de software há pelo menos 1 ano, e tiveram contato

Tabela 5.1: Comparativo entre as duas abordagens

Área	Planilha	Ferramenta
Mecânica de Avaliação	Já é um padrão bem difundido entre os avaliadores, e a forma de preenchimento dos dados é feita utilizando-se um software comum do dia-a-dia da organização.	Apesar de ser desenvolvida para ser algo simples e de fácil aprendizado, é uma nova abordagem. Isto pode trazer um pouco de receio quanto ao uso e a confiabilidade.
Inserção dos Resultados e Evidências	Os dados são inseridos de forma textual, onde os valores são definidos, mas não restritos às opções fornecidas, podendo ser alterados. Isto pode gerar erros	Os dados são definidos e limitados por quem gerou a ferramenta, através do método de avaliação. Então, a inserção de dados na ferramenta se limita a um conjunto limitado de opções, de acordo com a modelagem.
Inserção de Novos Dados	A inserção de novos dados na planilha (Projetos, Processos, Atributos, etc) é feita através da inserção de linhas e colunas na mesma. Isto pode comprometer a organização e acesso aos dados, bem como a mecânica geral da planilha.	Assim como na área anterior, o uso da ferramenta é definido pelos modelos de dados como entrada na engine geradora. Ou seja, uma vez entendida, a mecânica da ferramenta não irá ser alterada por inserções de novos dados na planilha
Acesso aos dados	Os dados da avaliação estão armazenados na própria planilha. Ou seja, tendo a planilha para avaliação, tem-se os dados da mesma.	Os dados estão armazenados na interface web da ferramenta, sendo acessível a qualquer momento e de qualquer lugar conectado à internet.
Compartilhamento dos dados	O compartilhamento dos dados é feito através do compartilhamento da própria planilha. Sendo assim, a cópia do arquivo pode escalar de forma que quem realizou a avaliação perca o controle dos dados	Os dados podem ser acessados via web. Além disso, um relatório estático pode ser emitido para envio em formato de arquivo, não permitindo edição dos dados.

com ferramentas de avaliação. Isso inclui gerentes de projeto, gerentes de configuração, analistas de qualidade, avaliadores, entre outros.

O questionário (mostrado no Apêndice B) foi enviado via email ou passado impresso após apresentação presencial da ferramenta. Quando enviado por email, um manual de uso (que está no Apêndice A) foi enviado também, com o intuito de instruir o profissional sobre o

Tabela 5.2: Comparativo entre as ferramentas de avaliação

Ferramenta	Plataforma	Aderente ao CMMI	Aderente ao MPS.BR	Aderente a ISO 15505	Suporta todas as Etapas de Avaliação	Adequação a mudanças na norma
ProEvaluator	Desktop	Não	Sim	Não	Não	Nenhuma
ProEvaluator 2.0	Web	Sim	Sim	Sim	Não	Nenhuma
FAPS	Desktop	Sim	Sim	Sim	Não	Nenhuma
Engine	Web	Sim	Sim	Sim	Não	Total

funcionamento básico da ferramenta.

Em ambos os casos (avaliação presencial ou por email), todos os profissionais que avaliaram a ferramenta utilizaram os mesmos artefatos de entrada (metamodelo, modelo e regras de avaliação), baseados no MPS.BR e MA-MPS.

É importante frisar que todos os usuários responderam ao questionário após a utilização fim-a-fim da solução, tanto da engine quanto da ferramenta gerada.

5.5.2 Resultados Obtidos

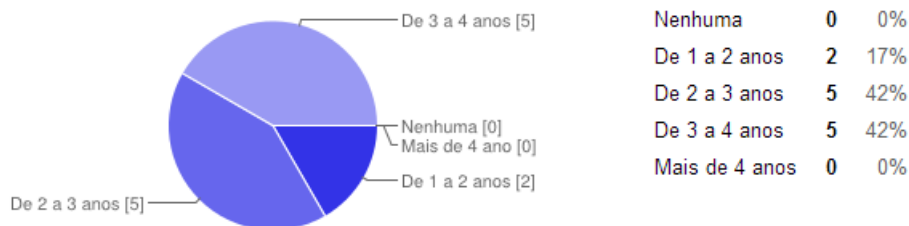
5.5.2.1 Questões Objetivas

As respostas dos entrevistados mostra que tanto a engine quanto a ferramenta estão com níveis de usabilidade considerado bom ou muito bom pela maioria deles. O fato do profissional ter utilizado a solução em todas as suas fases habilitou-o a dar uma opinião geral sobre a ferramenta, em vez de limitar esta opinião a cada ponto do sistema.

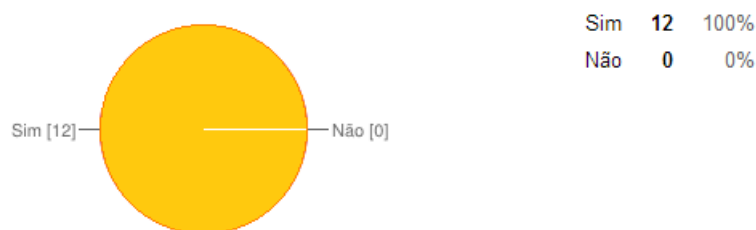
As perguntas foram divididas em três grupos com objetivos distintos:

- **Experiência:** O primeiro teve como objetivo verificar e medir a experiência do profissional em avaliações de processo de software.
- **Engine:** Teve como objetivo avaliar a engine de geração, visando analisar sua usabilidade e como é o gerenciamento de artefatos.
- **Ferramenta Gerada:** Teve como objetivo avaliar a ferramenta gerada. Como foi sentida, em termos de experiência de uso, a utilização da ferramenta durante uma avaliação.

A Figura 5.10 mostra a experiência dos entrevistados com avaliação de processos de software, medido em anos.

Figura 5.10: Experiência em Avaliação de Processos**Qual a sua experiência com avaliação de processos?****Fonte: O Autor**

A Figura 5.11 mostra a experiência dos entrevistados com implantação de melhoria de processos em organizações, de acordo com os critérios definidos na Seção 5.5.1. Todos os entrevistados tiveram contato com avaliação em uma ou mais fases do método de avaliação.

Figura 5.11: Trabalho com Implantação de melhoria de procesos**Já trabalhou em alguma organização com a implantação de melhoria de processos?****Fonte: O Autor**

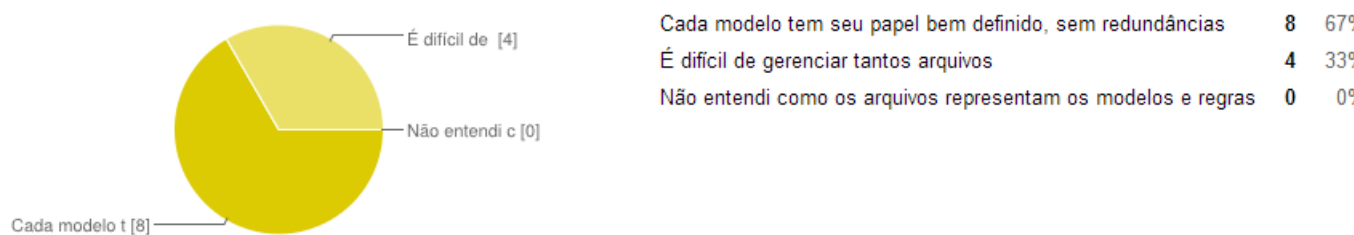
A Figura 5.12 mostra quais dos entrevistados teve contato com alguma ferramenta de avaliação de processo de software. No caso, é visto que todos tiveram contato, o que se mostra fundamental para uma análise comparativa da ferramenta com outras previamente utilizada.

Figura 5.12: Experiência com Ferramentas de Avaliação**Conhece ou já utilizou alguma ferramenta de avaliação de processo?****Fonte: O Autor**

A Figura 5.13 inicia as perguntas relativas à Engine, ou seja, relativas à utilização dos modelos e gerenciamento das ferramentas já geradas. Na primeira, os entrevistados são questionados sobre o fato de existir mais de um artefato de entrada e como isso pode afetar a experiência de uso.

Figura 5.13: Opinião sobre arquivos diferentes para uma única norma/modelo

Em relação ao uso de diferentes modelos/normas por meio de arquivos separados

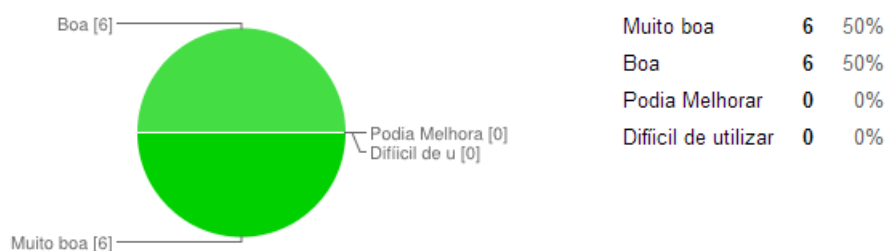


Fonte: O Autor

A Figura 5.14 é relativa à interface da Engine em si. Ela tem como objetivo medir a sensação do usuário ao utilizar a Engine para gerar a ferramenta de avaliação.

Figura 5.14: Opinião Sobre a Interface da Engine

O que achou da interface para geração das ferramentas?

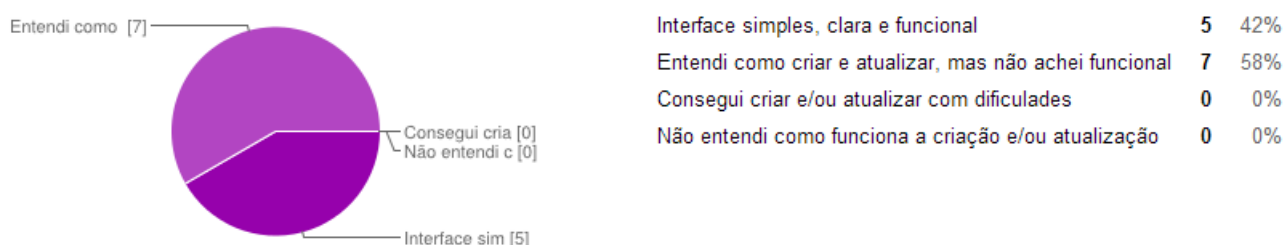


Fonte: O Autor

A Figura 5.15 é relativa à geração da ferramenta de avaliação através da Engine. Ela tem como objetivo avaliar a inserção dos dados da organização e da avaliação a ser realizada na ferramenta. Além disso, visa avaliar como a utilização dos dados contidos nos artefatos é percebida, de maneira geral, pelo usuário.

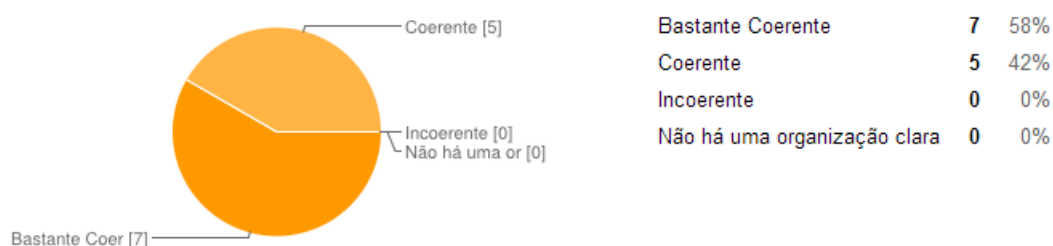
Figura 5.15: Opinião sobre a mecânica de geração da ferramenta

O que achou da mecânica de criação e atualização de ferramentas?

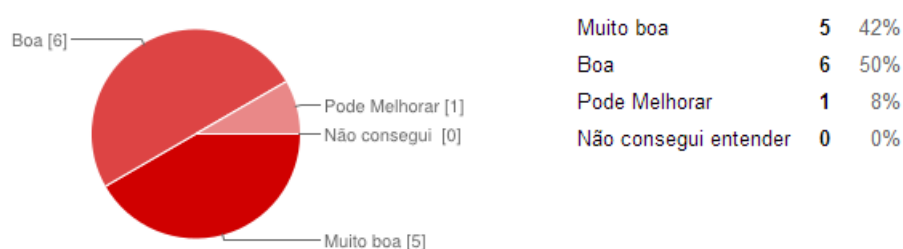


Fonte: O Autor

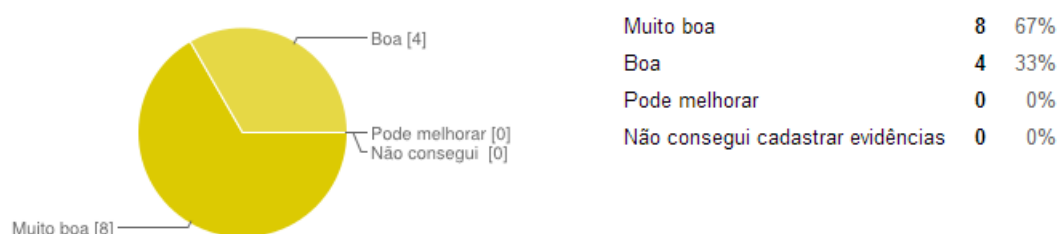
A Figura 5.16 inicia as perguntas relativas às ferramentas geradas pela Engine. Sua interface, a organização dos elementos, os cadastros auxiliares (projetos e evidências), bem como os mecanismos de persistência e sumarização dos dados são vistos pelos entrevistados. Inicialmente, os entrevistados são questionados sobre como a organização dos dados na ferramenta gerada.

Figura 5.16: Opinião sobre organização dos dados na ferramenta**Como você avalia a organização dos dados da ferramenta?**

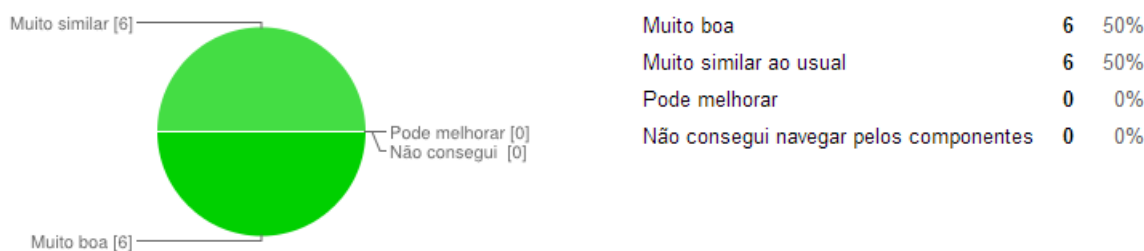
A Figura 5.17 mostra a resposta dos entrevistados quando questionados sobre como os projetos da organização são inseridos na ferramenta. As ferramentas precisaram ser geradas efetivamente para que esta pergunta pudesse ser respondida.

Figura 5.17: Opinião sobre o relacionamento de projetos da organização à avaliação**Como você avalia a forma de relacionar os projetos da organização à ferramenta?****Fonte:** O Autor

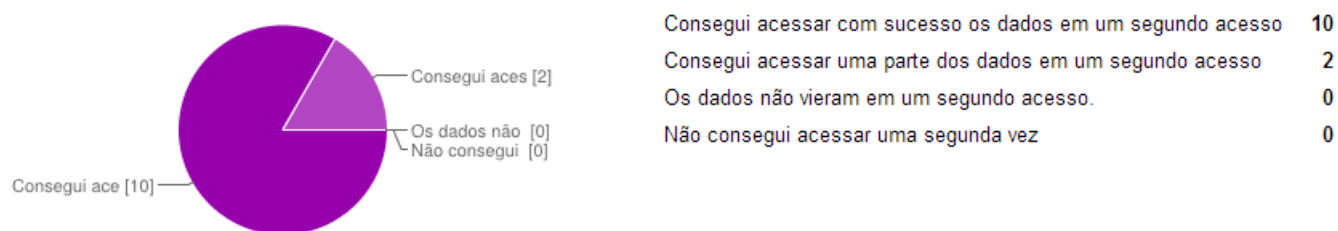
A Figura 5.18 mostra a resposta dos entrevistados quando questionados sobre como as evidências são relacionadas aos projetos da organização, em cada um dos componentes a serem avaliados do processo. Esta pergunta avalia como o relacionamento dos cadastros auxiliares é feito pela ferramenta, e se essa ação é bem aceita pelo usuário.

Figura 5.18: Opinião sobre o relacionamento de evidências aos projetos**Como você avalia a forma de relacionar as evidências aos projetos?****Fonte:** O Autor

A Figura 5.19 mostra a resposta dos entrevistados quando questionados sobre a forma de inserir a graduação de cada componente do processo. De acordo com as informações fornecidas nos modelos, as graduações são colocadas como uma lista, sem possibilidade de edição.

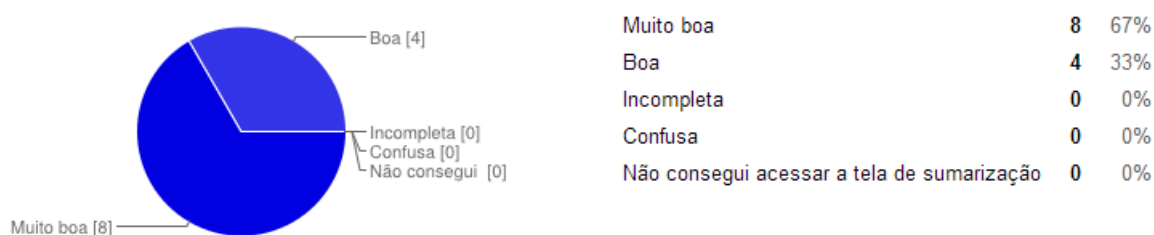
Figura 5.19: Opinião sobre a graduação dos componentes do processo**Como você avalia a forma de inserir as graduações para os componentes do processo****Fonte:** O Autor

A Figura 5.20 mostra a opinião dos entrevistados sobre como a persistência dos dados funcionou durante a utilização da ferramenta. Ou seja, se ao fechar o browser e acessá-lo novamente, os dados previamente inseridos são exibidos novamente para o usuário.

Figura 5.20: Opinião sobre a persistência dos dados na ferramenta**Como você avalia a persistência dos dados na ferramenta?****Fonte:** O Autor

A Figura 5.21 mostra a opinião dos entrevistados sobre a sumarização dos dados após a submissão dos mesmos e as regras do método de avaliação serem aplicadas nos dados submetidos na ferramenta. Esta pergunta serve para o usuário avaliar o tratamento dos dados inseridos e se eles são exibidos de maneira correta e coerente depois da execução da ferramenta.

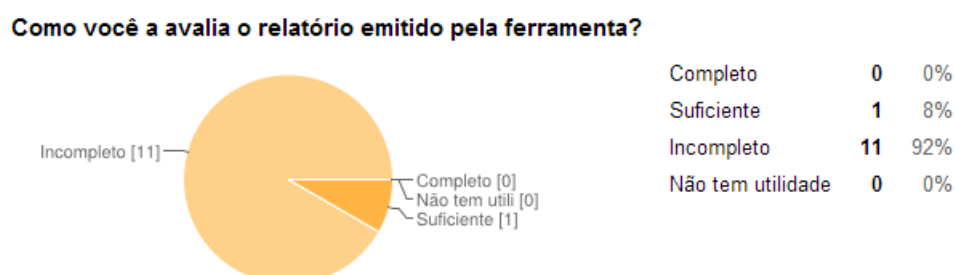
É uma forma também do entrevistado avaliar como as regras e modelos fornecidos são combinados automaticamente para gerar a avaliação do processo.

Figura 5.21: Opinião sobre a sumarização dos dados na ferramenta**Como você avalia a sumarização dos dados da ferramenta?****Fonte:** O Autor

A Figura 5.22 mostra a opinião dos entrevistados sobre o relatório emitido pela ferra-

menta, com os dados inseridos pelo usuário.

Figura 5.22: Opinião sobre o relatório emitido pela ferramenta



Fonte: O Autor

5.5.2.2 Questões Subjetivas

Pontos Fortes

Um ponto forte considerado por todos os entrevistados foi o fato da ferramenta ser gerada automaticamente, a partir de modelos. A ferramenta gerada foi considerada um ponto forte, já que por definição, independente do modelo ou norma fornecido como entrada, a interface é a mesma, considerada simples e funcional.

Pontos Fracos

O relatório foi considerado como o principal ponto fraco da solução, considerado básico demais, além de não permitir a customização de estilo. Além disso, a tela de inserção dos dados foi considerada inadequada se a quantidade de elementos for muito grande, além da falta de ajuda ao usuário durante a inserção dos dados.

O formato dos modelos utilizados como entrada também foi considerado um ponto fraco, dado que pode dificultar o entendimento de usuários pouco experientes.

Melhorias Sugeridas

Além do relatório, que foi considerado o ponto mais fraco a ser melhorado de forma a possibilitar customização, a tela de interface da ferramenta foi mencionada. A customização da tela, bem como a disponibilização de um tutorial ou interface de ajuda durante a avaliação também foram citados como melhorias.

Apesar de estar fora do escopo inicial da ferramenta, muitos dos entrevistados sugeriram que uma ferramenta de modelagem ou alguma outra forma de modelagem fosse incorporada à solução.

5.5.3 Análise dos Resultados

De maneira geral, os entrevistados consideraram válida a proposta da solução, de ser flexível tanto a diferentes normas e modelos de qualidade, e permitir que a atualização das ferramentas geradas pudesse ser feita através de atualizações nos artefatos de entrada.

A interface foi considerada satisfatória, dado que a mesma é gerada de forma automática. Mas mesmo com essa ressalva, a interface ainda precisa ser melhorada, além de permitir customização por parte dos usuários. Os mecanismos de persistência, atualização das ferramentas e acesso a elas também foram considerados satisfatórios.

O relatório emitido pela ferramenta foi um ponto bastante mencionado pelos entrevistados, que consideraram importante este ser mais elaborado e com melhor acabamento de estilo. Os artefatos utilizados, apesar de componentes da abordagem de desenvolvimento, foram questionados. Seu entendimento por parte de usuários menos experimentados foi colocado como um obstáculo a ser superado.

5.6 Considerações Finais

A ferramenta gerada pela engine é uma alternativa válida para avaliação, pois adiciona elementos de compartilhamento e disponibilidade dos dados ao contexto de avaliações. Dado que o compartilhamento da avaliação em curso se dá pelo compartilhamento do arquivo, a ferramenta fornece mais controle, evitando que o acesso a esses dados saia do controle da equipe de avaliação.

6

Conclusão

6.1 Contexto

A abordagem MDD visa diminuir o tempo de desenvolvimento de software, bem como prolongar a vida útil dos artefatos que compõem o processo de desenvolvimento (PASTOR et al., 2008). Por isso, se adequa às condições do mercado de software, cada vez mais competitivo e que demanda mais eficiência dos processos.

Outra demanda do mercado diz respeito à qualidade do produto a ser entregue. Nesse contexto, é comum atrelar a qualidade do produto à qualidade do processo que gera o mesmo (MOHAGHEGHI; AAGEDAL, 2007). Sendo assim, um processo com elevado nível de qualidade leva a um produto de qualidade, onde esta é medida de acordo com padrões de qualidade.

Com o alcance do mercado atingiu níveis globais, é de se esperar que as empresas que estejam nesse competitivo mercado se adequem a normas e padrões internacionais de qualidade para software. As ISO 15504 (ISO, 2004a;ISO, 2003;ISO, 2004b;ISO, 2004c;ISO, 2006), O CMMI (SEI, 2010), e ainda o MPS.BR (SOFTEX, 2013a) (com foco no processo de software brasileiro, mas que se adequa também aos padrões internacionais), figuram como os padrões utilizados em diversas organizações.

Mas a aplicação desses padrões tem que ser medidas baseando-se em um meio em que o processo possa ser avaliado e graduado em níveis, de acordo com a maturidade e capacidade do processo em garantir a qualidade de um produto MOHAGHEGHI; DEHLEN; NEPLE, 2009. Por isso, cada norma possui o seu chamado método de avaliação.

O SCAMPI (SEI, 2011) é o método de avaliação do CMMI, e o MA-MPS, o método de avaliação do MPS.BR definem como um processo deve ser avaliado de acordo com cada uma das normas. Definem ainda o que cada um dos seus níveis representa. Além disso, contêm meios de atingirem os níveis gradualmente, dado que as avaliações acontecem várias vezes na mesma organização. Ou seja, uma avaliação por um método de qualidade pode ser aplicada em diferentes períodos de tempo, com diferentes áreas a serem avaliadas. Essa avaliação se estende, inclusive, a áreas previamente avaliadas, porém com critérios diferentes, dado que os níveis de exigência vão se tornando mais altos com o tempo.

Diante do contexto de eficiência e agilidade em se ter um produto, uma avaliação de processo não pode ter um custo elevado tanto financeiramente, quanto de tempo. Para isso, muitas pesquisas e trabalhos foram feitos no sentido de organizar e automatizar ambientes para avaliação de processos de software (OLIVEIRA; VASCONCELOS; ROUILLER, 2005; XAVIER, 2007; SOTERO, 2009; THIRY et al., 2008).

Muitos dos trabalhos renderam ferramentas que têm este objetivo de prover apoio automatizado à avaliação de processos, baseando-se nas formas usuais de avaliação. As ferramentas, porém, em sua maioria, são concebidas para que atendam a maior quantidade de empresas da mesma maneira. Isso tem suas vantagens, mas há desvantagens associadas.

Uma das desvantagens é que qualquer alteração que necessite ser feita na ferramenta (devido à mudanças na norma ou no método de avaliação, por exemplo), terá que ser escalada a um nível muito grande, que geraria custo tanto para a empresa que desenvolveu a ferramenta, quanto para quem utiliza a ferramenta.

Então, este trabalho se propôs a apresentar uma solução a estes problemas de escalabilidade de mudanças, desenvolvimento ágil e personalizado de ferramentas pelas próprias organizações para avaliação de processos de software.

6.2 Contribuições do Trabalho

As principais contribuições do trabalho estão na capacidade de gerar automaticamente ferramentas baseadas em modelos e na sua disponibilidade ininterrupta para avaliação e armazenamento de dados. Por estar em um ambiente web, a ferramenta se propõe a estar disponível sempre que o usuário necessitar.

Num contexto onde a ferramenta define a forma de avaliar e a norma em si, ou seja, que dados e de que forma os dados coletados serão armazenados, uma engine onde o modelo definido pelo usuário é suficiente para resultar numa ferramenta é flexível o bastante para atender organizações com diversos níveis de maturidade e de capacidade.

Mesmo com uma proposta diferente de desenvolvimento de ferramentas, a interface das mesmas foi desenvolvida para que se assemelhasse aos métodos atuais de avaliação utilizados. Somado a isso, vários conceitos de interface web foram utilizados para fornecer uma experiência que se propõe a ser agradável aos usuários, tanto para quem irá gerar as ferramentas, quanto para quem irá utilizar a ferramenta gerada.

A forma de atualização da ferramenta também é uma contribuição importante, dado que o usuário é responsável pela mesma, com a engine e ferramentas se adequando ao modelo que foi criado pelo usuário.

6.3 Trabalhos Futuros

Durante o desenvolvimento do trabalho, várias oportunidades de melhoria foram identificadas, mas, por não se encaixarem no escopo inicial dos requisitos do capítulo 4, foram agrupadas para serem apresentadas como propostas de extensão. Isso se deu pelo fato do escopo da atividade de Execução da Avaliação dos métodos de avaliação estudados ter sido escolhido, bem como pelo fato de que a modelagem das normas e métodos foi deixada como externa a ferramenta contribuíram para essa restrição no escopo.

- **Adequação à plataformas mobile:** Esta melhoria acrescentaria mobilidade às ferramentas geradas, além de aproximar ainda mais o usuário da engine de geração de ferramentas. Sendo assim, o usuário não estaria restrito apenas à uma interface de computador, mas também ao seu smartphone ou tablet.
- **Integração com ferramentas de modelagem:** A modelagem dos artefatos de entrada não estava considerada no escopo deste trabalho. Porém, a possibilidade dos modelos serem criados de maneira visual pelos usuários que irão gerar as ferramentas de avaliação tornaria o ambiente ainda mais robusto.
- **Envio de relatórios por email:** Adição de um conjunto de emails para serem comunicados cada vez que resultados de avaliações sejam submetidos à ferramenta gerada. Isso tornaria a comunicação entre a equipe de avaliação mais rápida e ágil, onde todos teriam conhecimento, em tempo real, das submissões de dados à ferramenta.
- **Validação da solução em ambiente real:** Apesar de validada com uma avaliação real, a ferramenta ainda carece de testes, principalmente testes alfa e beta, para sua usabilidade ser medida em ambiente real de utilização.
- **Mais de um tipo de relatório:** Atualmente tendo apenas a opção de emitir relatórios em PDF com um formato próprio, a engine poderia fornecer meios de construção de relatórios customizados. Ou seja, o usuário poderia escolher que dados seriam exibidos, a forma de exibição (fonte, posicionamento, etc), bem como caracterizar o relatório de acordo com a organização avaliada, por exemplo.
- **Implementação de um mecanismo de autenticação:** Como está em fase experimental, a ferramenta não tem distinção de usuários. O acesso é feito via URL, sem validação ou restrição.

Referências

- ATKINSON, C.; KÜHNE, T. Model-Driven Development: a metamodeling foundation. **EEE Software**, [S.l.], v.20, n.5, p.36–41., 2003.
- CHEN, N.; HOI, S.; XIAO, X. Software Process Evaluation: a machine learning approach. **International Conference on Automated Software Engineering**, [S.l.], v.26, November 6-12 2011.
- FOUNDATION, E. **Eclipse Modeling Framework Project (EMF)**. 2010.
- GITZEL, R.; KORTHAUS, A. The Role of Metamodeling in Model-Driven Development. In: **WORLD MULTI-CONFERENCE ON SYSTEMICS, CYBERNETICS AND INFORMATICS. Proceedings...** [S.l.: s.n.], 2004.
- GROUP, P. **PHP**. 2001.
- ISO. **ISO/IEC 12207**: information technology – software life cycle processes. 1995.
- ISO. **ISO/IEC 15504-2**: information technology - process assessment - part 2 - performing an assessment. 2003.
- ISO. **ISO/IEC 15504-1**: information technology - process assessment - part 1 - concepts and vocabulary. 2004.
- ISO. **ISO/IEC 15504-3**: information technology - process assessment - part 3 - guidance on performing an assessment. 2004.
- ISO. **ISO/IEC 15504-4**: information technology - process assessment - part 4 - guidance on use for process improvement and process capability determination. 2004.
- ISO. **ISO/IEC 15504-5**: information technology - process assessment - part 5: an exemplar process assessment model. 2006.
- MCFEELEY, R. **IDEAL**: a user's guide for software process improvement. 1996.
- MOHAGHEGHI, P.; AAGEDAL, J. **Evaluating Quality in Model-Driven Engineering**. 2007.
- MOHAGHEGHI, P.; DEHLEN, V. A Metamodel for Specifying Quality Models in Model-Driven Engineering. **Nordic Workshop on Model Driven Engineering**, [S.l.], 2008.
- MOHAGHEGHI, P.; DEHLEN, V.; NEPLE, T. Definitions and approaches to model quality in model-based software development - A review of literature. **Information and Software Technology**, [S.l.], v.v.51, n.n.12,, p.p.1646–1669, December 2009.
- NBR/ISO. **Sistema de Gestão da Qualidade – Fundamentos e Vocabulário - ISO 9000**. 2000.
- OLIVEIRA, S.; VASCONCELOS, A.; ROUILLER, A. Uma Proposta de um Ambiente de Implementação de Processo de Software. **Revista InfoComp - Revista de Ciência da Computação da UFLA**, [S.l.], 2005.

- OMG. **Model-Driven Architecture.**, <<http://www.omg.org/mda/>> Acesso em: 18 nov. 2013.
- OMG. **XML Metadata Interchange (XMI).**, <<http://www.omg.org/spec/XMI/2.4.2>> Acesso em: 14 abr. 2013.
- OMG. **Object Constraint Language (OCL).**, <<http://www.omg.org/spec/OCL/2.4>> Acesso em: 24 fev. 2013.
- OMG. **MetaObject Facility - MOF.**, <<http://www.omg.org/mof/>> Acesso em: 18 nov. 2013.
- PASTOR, O. et al. Model-driven development. **Inform Spektrum**, [S.l.], v.31(, n.5, p.394–407, 2008.
- ROUT, P. SPICE: a framework for software process assessment. **Software Process – Improvement and Practice.**, [S.l.], 1995.
- SEI. **CMMI for Development, Version 1.3 - CMMI-DEV, V1.3.**, <<http://www.sei.cmu.edu/reports/10tr033.pdf>> Acesso em: 02 mar. 2014.
- SEI. **Appraisal Requirements for CMMI Version 1.3 (ARC, V1.3).** [S.l.]: Software Engineering Institute, 2011.
- SENDALL, S.; KOZACZYNSKI, W. Model transformation - The heart and Soul of Model driven Development. **IEEE Software**, [S.l.], p.42, September 2003.
- SOFTEX. **Guia Geral de Software.**, <http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_2012.pdf> Acesso em: 02 mar. 2014.
- SOFTEX. **Guia de Avaliação.**, <http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_de_Avaliacao_2013.1.pdf> Acesso em: 09 mar. 2013.
- SOFTEX. **Guia de Implementação de Software - Parte 2: nível f.**, <http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_de_Implementacao_Parte_2_20131.pdf> Acesso em: 19 set. 2013.
- SOFTEX. **Guia de Implementação de Software - Parte 2: nível a.**, <http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_de_Implementacao_Parte_7_2013.pdf> Acesso em: 19 set. 2013.
- SOTERO, T. **ProEvaluator uma ferramenta de apoio à avaliação de processos de software com foco em modelos de maturidade.** 2009. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Pernambuco.
- SUN, C. et al. Mining explicit rules for software process evaluation. In: INTERNATIONAL CONFERENCE ON SOFTWARE AND SYSTEM PROCESS (ICSSP 2013), 2013. **Proceedings...** [S.l.: s.n.], 2013. p.118–125.
- THIRY, M. et al. **FAPS Ferramenta para apoiar Avaliações Integradas de Processos de Software.** 2008.
- VASCONCELOS, A. M. L. et al. **Towards CMMI-compliant MDD Software Processes.** 2011.

XAVIER, J. **Proevaluator Uma ferramenta para avaliação de processos de software**. 2007. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Pernambuco.



Apêndice A: Manual de Uso

Este apêndice visa apresentar o manual de uso fornecido como componente da avaliação da solução pelos usuários entrevistados. Foi concebido para ser simples e direto, apenas se atendo aos elementos que devem ser levados em conta pelos usuários da ferramenta.

1. Objetivo do Documento

O presente documento tem como objetivo fornecer um guia de utilização para a engine de geração de ferramentas, criada como componente do projeto de Dissertação de Mestrado Acadêmico em Ciência da Computação do aluno Luiz Felipe de Oliveira Libório, sob orientação do Prof. Alexandre Vasconcelos, no Centro de Informática da Universidade Federal de Pernambuco.

2. Resumo do Trabalho

Atualmente, é cada vez mais exigido como critério competitivo a aderência do processo de software de uma organização a um modelo ou norma de qualidade. Alcançar altos níveis de aderência é uma tarefa complexa para as organizações, pois a atividade de avaliar essa aderência exige cuidado ao ser realizada. Com isso, busca-se cada vez mais o apoio de ferramentas de avaliação computadorizadas, cujo objetivo é agilizar e tornar mais confiável o processo de avaliação dos processos de software.

Ainda assim, um dos fatores que tendem a dificultar a utilização e propagação do uso de ferramentas é a mudança das normas de qualidade e dos métodos de avaliação associados, que visam se adequar constantemente às boas práticas do mercado. Ora, se uma ferramenta é criada baseada na norma de qualidade e no seu método de avaliação, qualquer mudança em algum dos dois componentes significa mudanças na ferramenta, gerando custos de manutenção e de distribuição da nova versão da ferramenta. Então, uma ferramenta que fosse sensível a mudanças das normas de avaliação, porém sem que seu processo de desenvolvimento fosse executado novamente encaixaria perfeitamente no contexto.

Ferramentas criadas através de processos MDD (Model-Driven Development) atendem a esses requisitos, dado que o seu objetivo é transformar elementos mais abstratos em elementos mais concretos utilizando uma sucessão de transformações, através de modelos. Aplicando as regras de MDD aos modelos de qualidade e métodos de avaliação, é possível a criação de ferramentas para avaliação baseada nos modelos que compõem a avaliação. Ou seja, através de um modelo de processo e de um metamodelo da norma de qualidade, além do método de avaliação, é possível criar de forma automática via MDD uma ferramenta que possa ser atualizada sob demanda, sem que seu código-fonte precise ser alterado manualmente.

Palavras-chave:

Normas de Qualidade, Métodos de Avaliação, Avaliação de Processos de Software, MDD, Model-Driven Development

3. Utilização da Engine

Para utilizar a Engine, o usuário deve acessar o endereço abaixo:

<http://www.lfliborio.com.br/msc/engine>

Após o carregamento da página, a seguinte página deve ser exibida, com os campos que precisam ser preenchidos pelo usuário marcados em vermelho:

The screenshot shows the 'Início' (Start) page of the MSc Engine. It features a navigation bar with 'Criar', 'Atualizar', and 'Contato' links. The main content is divided into three sections: 'Informações sobre a ferramenta' (Tool Information), 'Informações sobre a norma' (Standard Information), and 'Regras de Avaliação' (Evaluation Rules). Each section contains input fields or buttons that are highlighted with red boxes and numbered 1 through 8.

Início

Criar Atualizar Contato **8**

Informações sobre a ferramenta

1 Identificador:

2 Permalink:

3 Versão do metamodelo:

Informações sobre a norma

4 Seleccione o metamodelo

Metamodelo do processo, de acordo com a norma de qualidade sob a qual vai ser aplicada a avaliação. Deve possuir a extensão .xmi

5 Seleccione o modelo

Modelo da norma de qualidade a ser aplicada à avaliação, de acordo com o metamodelo informado anteriormente. Deve possuir a extensão .xmi

Regras de Avaliação

6 Seleccione o arquivo de regras

Regras de avaliação que serão aplicadas ao modelo para geração do resultado; Devem possuir a extensão .ocl

7 Gerar

1. Representa o nome da ferramenta, ou ainda, um identificador da avaliação a qual a ferramenta a ser gerada se destina.
2. Representa o nome da ferramenta traduzido em formato de url, para ser utilizado posteriormente para acesso da mesma. **Recomenda-se que o usuário guarde este valor para posterior consulta da ferramenta.**
3. Representa a versão do metamodelo (Norma de Qualidade) a ser usada na ferramenta. Tem caráter informativo, funcionando apenas como dado adicional à ferramenta.
4. Recebe o arquivo de metamodelo (Norma de Qualidade) cuja aderência do processo vai ser avaliada. O arquivo tem formato **.xmi**.

5. Recebe o arquivo de modelo do processo cuja aderência do processo vai ser avaliada. O arquivo tem formato **.xmi**.
6. Recebe o arquivo das regras que regem a avaliação do processo. O arquivo tem formato **.ocl**.

As entradas criadas para serem utilizadas como exemplo estão disponíveis [aqui](#).

7. Submete os dados informados para gerar a ferramenta desejada.
8. Caracteriza o menu da engine, que possibilita a criação de uma ferramenta nova, a atualização de uma ferramenta anteriormente criada e o contato com o desenvolvedor da ferramenta para feedbacks.

4. Utilização da Ferramenta Gerada

4.1. Tela de Inserção de Projetos

Após a entrada dos modelos e dados da organização, é preciso cadastrar dados relacionados à avaliação. Esses dados são inseridos na tela abaixo, ao qual o usuário é direcionado diretamente:

Avaliação - MPS.BR

▲ Informações sobre a avaliação

Avaliador: 1

Data de Início: 2

Data de Término: 3

Projeto: 4 5

Projeto	Excluir
P1	✖ 6
P2	✖

7

1. Este campo destina-se a receber o nome do responsável pela avaliação;
2. Este campo representa a data em que a avaliação se inicia;
3. Este campo representa a data em que a avaliação está prevista para terminar;
4. Neste campo, o usuário insere o nome do projeto a ser inserido no contexto da avaliação
5. Adiciona o dado inserido no campo 4 na tabela.
6. Remove o projeto correspondente da tabela.
7. Submete os dados informados para prosseguir para a avaliação.

4.3. Tela de Inserção de Evidências

Avaliação - MPS BR

1. Evidência:

2. Fonte:

3. Adicionar Projeto: +

4. Projetos: P2

5. Adicionar Evidência +

Evidência	Fonte	Projetos	Excluir
El - Evidência 1	Fonte 1	P1, P2	

6. Excluir

1. Título da Evidência a ser inserida;
2. Fonte da Evidência a ser inserida;
3. Lista com os projetos na qual a evidência foi encontrada, que podem ou não ser inseridos na evidência;
4. Projetos já selecionados;
5. Botão que cadastra a evidência na tabela;
6. Botão que permite remover uma evidência já cadastrada.

B

Apêndice B: Questionário

Este apêndice visa apresentar o questionário que foi utilizado para avaliação da solução pelos usuários. Foi concebido para cobrir pontos referentes à engine, às ferramentas geradas e à usabilidade geral da solução.

21/7/2014

Avaliação de Engine para Geração de Ferramentas de Avaliação - Google Forms

Sobre o avaliador

* Required

1. Nome:

.....

2. Email

.....

3. Qual a sua experiência com avaliação de processos? *

Atividades de auditoria interna, consultoria ou avaliação propriamente dita

Mark only one oval.

- ☐ Nenhuma
- ☐ De 1 a 2 anos
- ☐ De 2 a 3 anos
- ☐ De 3 a 4 anos
- ☐ Mais de 4 anos

4. Já trabalhou em alguma organização com a implantação de melhoria de processos? *

Utilizou CMMI, [MPS.BR](#), etc, tanto implantando melhoria ou trabalhando em organização que teve melhoria implantada.

Mark only one oval.

- ☐ Sim
- ☐ Não

5. Conhece ou já utilizou alguma ferramenta de avaliação de processo? *

Mark only one oval.

- ☐ Sim
- ☐ Não

6. Se você respondeu que sim à questão anterior, informe qual(is) ferramenta(s) foi(ram) utilizada(s):

.....

.....

.....

.....

.....

Sobre a engine

21/7/2014

Avaliação de Engine para Geração de Ferramentas de Avaliação - Google Forms

7. Em relação ao uso de diferentes modelos/normas por meio de arquivos separados *

Depois de analisar os artefatos fornecidos como exemplo, qual a avaliação do entrevistado sobre a abordagem?

Mark only one oval.

- ☐ Cada modelo tem seu papel bem definido, sem redundâncias
- ☐ É difícil de gerenciar tantos arquivos
- ☐ Não entendi como os arquivos representam os modelos e regras

8. O que achou da interface para geração das ferramentas?

Ao analisar a organização dos campos, textos de ajuda, validações, qual a avaliação do entrevistado sobre a abordagem?

Mark only one oval.

- ☐ Muito boa
- ☐ Boa
- ☐ Podia Melhorar
- ☐ Difícil de utilizar

9. O que achou da mecânica de criação e atualização de ferramentas?

Saber qual dos dois menus utilizar, quando e como localizar as ferramentas já geradas

Mark only one oval.

- ☐ Interface simples, clara e funcional
- ☐ Entendi como criar e atualizar, mas não achei funcional
- ☐ Consegui criar e/ou atualizar com dificuldades
- ☐ Não entendi como funciona a criação e/ou atualização

10. O manual de uso consegue transmitir de maneira clara como a engine funciona e como as ferramentas são geradas?

Mark only one oval.

- ☐ Completamente
- ☐ Parcialmente
- ☐ Não consegui entender as instruções

Sobre a ferramenta gerada**11. Como você avalia a organização dos dados da ferramenta?**

A organização em áreas minimizáveis está adequada e agrupa informações coesas?

Mark only one oval.

- ☐ Bastante Coerente
- ☐ Coerente
- ☐ Incoerente
- ☐ Não há uma organização clara

21/7/2014

Avaliação de Engine para Geração de Ferramentas de Avaliação - Google Forms

12. Como você avalia a forma de relacionar os projetos da organização à ferramenta?

O fato dessa atividade acontecer antes da entrada de dados altera a experiência de uso de alguma maneira?

Mark only one oval.

- ☐ Muito boa
- ☐ Boa
- ☐ Pode Melhorar
- ☐ Não consegui entender

13. Como você avalia a forma de relacionar as evidências aos projetos?

Mark only one oval.

- ☐ Muito boa
- ☐ Boa
- ☐ Pode melhorar
- ☐ Não consegui cadastrar evidências

14. Como você avalia a forma de inserir as graduações para os componentes do processo

Mark only one oval.

- ☐ Muito boa
- ☐ Muito similar ao usual
- ☐ Pode melhorar
- ☐ Não consegui navegar pelos componentes

15. Como você avalia a persistência dos dados na ferramenta?

Considere um segundo acesso como um acesso ao link diretamente, sem gerar novamente a ferramenta.

Mark only one oval.

- ☐ Consegui acessar com sucesso os dados em um segundo acesso
- ☐ Consegui acessar uma parte dos dados em um segundo acesso
- ☐ Os dados não vieram em um segundo acesso.
- ☐ Não consegui acessar uma segunda vez

16. Como você avalia a sumarização dos dados da ferramenta?

Para responder a esta pergunta, você precisa inserir e submeter dados para avaliação.

Mark only one oval.

- ☐ Muito boa
- ☐ Boa
- ☐ Incompleta
- ☐ Confusa
- ☐ Não consegui acessar a tela de sumarização

21/7/2014

Avaliação de Engine para Geração de Ferramentas de Avaliação - Google Forms

17. Como você avalia o relatório emitido pela ferramenta?

Para responder a esta pergunta, você precisa emitir o relatório

Mark only one oval.

- ☐ Completo
- ☐ Suficiente
- ☐ Incompleto
- ☐ Não tem utilidade

Pontos Fortes, Pontos Fracos, Melhorias

Nesta páginas, você tem espaço para escrever livremente suas impressões sobre o objeto da avaliação.

18. Pontos Fortes

Quais pontos fortes foram identificados na ferramenta e/ou engine?

19. Pontos Fracos

Quais pontos fracos foram identificados na ferramenta e/ou engine?

20. Melhorias Sugeridas

Em que a engine e/ou ferramentas podem melhorar?

Powered by
 Google Forms