

UNIVERSIDADE FEDERAL DO ESTADO DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E
SISTEMAS

**UMA ANÁLISE DO PROCESSO DE DESENVOLVIMENTO DE
UM REPOSITÓRIO DE CONTEÚDO PARA EDUCAÇÃO DE
TRÂNSITO BASEADA NO MODELO DE REFERÊNCIA MPS
PARA SOFTWARE**

Leonardo Sgnaolin

FLORIANÓPOLIS
2019

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E
SISTEMAS

Leonardo Sgnaolin

**UMA ANÁLISE DO PROCESSO DE DESENVOLVIMENTO DE
UM REPOSITÓRIO DE CONTEÚDO PARA EDUCAÇÃO DE
TRÂNSITO BASEADA NO MODELO DE REFERÊNCIA MPS
PARA SOFTWARE**

Trabalho de Conclusão de Curso
submetido ao Programa de Graduação
da Universidade Federal de Santa
Catarina para a obtenção do Grau de
bacharel em Engenharia de Produção
Civil
Orientador: Prof. Dr. Ricardo Villarroel
Dávalos

FLORIANÓPOLIS
2019

Ficha de identificação da obra elaborada pelo autor
através do Programa de Geração Automática da Biblioteca Universitária
da UFSC.

Sgnaolin, Leonardo

UMA ANÁLISE DO PROCESSO DE DESENVOLVIMENTO DE UM
REPOSITÓRIO DE CONTEÚDO PARA EDUCAÇÃO DE TRÂNSITO
BASEADA NO MODELO DE REFERÊNCIA MPS PARA SOFTWARE. /

Leonardo Sgnaolin; orientador, Ricardo Villarroel Dávalos, 2019.

150 p.

Trabalho de Conclusão de Curso (graduação) Universidade Federal de Santa
Catarina, Centro Tecnológico, Graduação em Engenharia de Produção Civil,
Florianópolis, 2019.

Inclui referências.

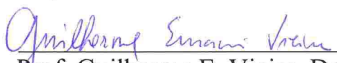
1. Engenharia de Produção Civil. 2. MPS.BR. 3. Avaliação. 4.
Desenvolvimento. 5. Software. I. Villarroel Dávalos, Ricardo. II. Universidade
Federal de Santa Catarina. Graduação em Engenharia de Produção Civil. III.
Título.

Leonardo Sgnaolin

**UMA ANÁLISE DO PROCESSO DE DESENVOLVIMENTO DE
UM REPOSITÓRIO DE CONTEÚDO PARA EDUCAÇÃO DE
TRÂNSITO BASEADA NO MODELO DE REFERÊNCIA MPS
PARA SOFTWARE.**

Este TCC foi julgado adequadamente e aprovado para obtenção do
Título de “Bacharel” e aprovado em sua forma final pelo Programa de
Graduação da Universidade Federal de Santa Catarina

Florianópolis, 16 de novembro de 2019.



Prof. Guilherme E. Vieira, Dr.

Coordenadora dos Cursos de Graduação em Engenharia de Produção

Banca Examinadora:



Prof. Ricardo Villarroel Dávalos, Dr.
Orientador

Universidade Federal de Santa Catarina



Prof.ª Olga Regina Cardoso, Dr.ª
Universidade Federal de Santa Catarina



Prof. Eduardo Ferreira da Silva, Dr.
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus
colegas de classe e à minha família.

AGRADECIMENTOS

Agradeço a meu pai, Luis Carlos Sgnaolin, minha mãe, Jociane Firmo, e a toda minha família que sempre me apoiaram e incentivaram no que fosse necessário. Um agradecimento especial a minha irmã, Francine Sgnaolin, que me apoiou na revisão deste trabalho.

Aos meus amigos e colegas de trabalho, que estiveram presentes, incentivaram e deram conselhos que contribuíram para a melhoria deste trabalho.

A empresa do estudo de caso, pela oportunidade de realizar meu trabalho com base na implantação do modelo de melhoria de processos MR-MPS-SW.

Ao orientador Ricardo Villarroel Dávalos, por me auxiliar e incentivar em todos os momentos do desenvolvimento deste trabalho.

Agradeço também a todas outras pessoas que de alguma forma contribuíram para que eu pudesse desenvolver este trabalho.

“Nós só podemos ver um pouco do futuro,
mas o suficiente para perceber que há muito a
fazer.”

Alan Turing, 1950.

RESUMO

O setor de desenvolvimento de Sistemas de Informação vem crescendo rapidamente, com muitas empresas trabalhando na produção de *softwares* para seus clientes. Para realizar essa tarefa com qualidade e de uma maneira mais previsível, atendendo às estimativas de custos e aos compromissos dos seus cronogramas, são despendidos esforços em melhoramento de processo de desenvolvimento de software. No entanto, os *frameworks* de processos de desenvolvimento de software mais estabelecido, como SPICE (*Software Process Improvement and Capability dEtermination*) e CMMI (*Capability Maturity Model Integration*), são modelos difíceis e onerosos de implantação, tornando inviável a aplicação para a boa parte das empresas.

Esse trabalho pretende identificar quais são as melhorias recomendadas no processo de desenvolvimento do software de um repositório de conteúdo para educação de trânsito. Para isso, baseou-se no Nível G do modelo MR-MPS-SW do programa MPS.BR (Melhoria de Processos de Software Brasileiro). Esse modelo é uma adaptação do CMMI adequado ao cenário nacional e melhor ajustado às empresas de menores portes. E também, pode ser implementado com menor esforço e custo.

O MR-MPS-SW possibilitou identificar pontos fracos no processo de software do estudo de caso. As inconsistências foram na avaliação dos processos GRE 2, GPR 4, GPR 5 GPR 6, GPR 12 e GPR 15 e em seus atributos de processo. Com esses problemas identificados, pôs-se a propor iniciativas tangíveis de melhoria de processos com base em práticas recomendadas e bem estabelecidas de engenharia de software. Alguns ajustes no fluxo de desenvolvimento foram feitos, realocando algumas atividades, como práticas de gerenciamento de riscos, reunião de *kick-off*, método Planning Poker, e ferramentas, como a Matriz SWOT e Gráfico *Burndown*. Aplicar essas propostas permitirá que a melhoria de processos se tornar parte integrante do desenvolvimento organizacional e cultural na empresa.

Palavras-chave: MPS.BR; MR-MPS-SW; Nível G; tecnologia da informação e comunicação; TI; avaliação; desenvolvimento; software.

ABSTRACT

The Information Systems development industry has been growing rapidly, with many companies working to produce software for their customers. To accomplish this task with quality and in a more predictable manner, meeting cost estimates and schedule commitments, efforts are made to improve the software development process. However, the most established software development process frameworks, such as Software Process Improvement and Capability Termination (SPICE) and Capability Maturity Model Integration (CMMI), are difficult and costly deployment models, making application unfeasible for most companies. This paper aims to identify the recommended improvements in the software development process of a traffic education content repository. For this, it was based on Level G of the MR-MPS-SW model of the MPS.BR (Brazilian Software Process Improvement) program. This model is an adaptation of CMMI appropriate to the national scenario and better suited to smaller companies. Also, it can be implemented with less effort and cost.

MR-MPS-SW made it possible to identify weaknesses in the case study software process. The inconsistencies were in the evaluation of GRE 2, GPR 4, GPR 5 GPR 6, GPR 12 and GPR 15 processes and their process attributes. With these issues identified, it is set out to propose tangible process improvement initiatives based on well-established software engineering best practices. Some adjustments to the development flow were made, reallocating some activities, like risk management practices, kick-off meeting, Planning Poker method, and tools such as the SWOT Matrix and Burndown Chart.

Applying these proposals will enable process improvement to become an integral part of organizational and cultural development in the company.

Keywords: MPS.BR; MR-MPS-SW; Level G; information and communication technology; IT; assessment; development; software.

LISTA DE FIGURAS

Figura 1 - Ciclos de vida tradicionais.....	32
Figura 2 - Ciclos de vida ágeis e extremos.....	34
Figura 3 - Estrutura de melhoria do processo de software.	39
Figura 4 - Níveis de maturidade do CMMI.....	44
Figura 5 - Comparação dos níveis de maturidade do MPS.BR com do CMMI.	46
Figura 6 - Componentes do Programa MPS.BR.	48
Figura 7 - Fluxo de desenvolvimento de um incremento.	58
Figura 8 – Etapas metodológicas do trabalho.....	67
Figura 9 - Planejamento da avaliação.....	68
Figura 10 - Etapa de iniciação do projeto.....	72
Figura 11 - Etapa de Levantamento de Requisitos do Sistema	74
Figura 12 - Etapa de desenvolvimento do sistema.	77
Figura 13 - Etapa de validação do sistema	78
Figura 14 - Cartas de Planning Poker.....	91
Figura 15 - Processo inicial com tarefa de decomposição em User Storys.	92
Figura 16 - Gráfico de Evolução Regressiva de Iteração (Burndown)..	94
Figura 17 - Matriz SWOT.	95
Figura 18 - Processo de iniciação com as alterações sugeridas para atender o GPR12.	97
Figura 19 - Análise 5W2H (parte 1/2).....	98
Figura 20 - Análise 5W2H (parte 2/2).....	99

LISTA DE TABELAS

Tabela 1 - Níveis de maturidade do MR-MPS-SW.....	50
Tabela 2 – Ferramenta 5W2H.	61
Tabela 3 – Escala para caracterização do grau de implementação de um resultado esperado do processo nos projetos/serviços/área... 63	
Tabela 4 – Regras para caracterizar o grau de implantação dos atributos do processo na unidade organizacional.....	64
Tabela 5 – Caracterização de atributos do processo para satisfazer aos níveis MR-MPS.....	64
Tabela 6 - Caracterização do grau de implementação de cada resultado esperado.	88
Tabela 7 - Caracterização do grau de implementação de cada RAP e AP.	89

LISTA DE ABREVIATURAS E SIGLAS

ABES - Associação Brasileira das Empresas de Software
ABNT - Associação Brasileira de Normas Técnicas
AMP - Avaliação e Melhoria do Processo Organizacional
AP - Atributos de Processo
CMMI - *Capability Maturity Model Integration* (Modelo de Capacidade e Maturidade Integrado)
CMMI *Services* - CMMI para Serviços
CMMI *Supplier Management* - CMMI para Gerenciamento de Fornecedores
CMMI-DEV - CMMI *Development* (CMMI para Desenvolvimento)
DFP - Definição do Processo Organizacional
DM - *Daily Meeting* (Reunião diária)
DRE - Desenvolvimento de Requisitos
DRU - Desenvolvimento para Reutilização
EAP - Estrutura Analítica do Projeto
Equipe Dev - Equipe de desenvolvimento (papel do Scrum)
GCO - Gerência de Configuração
GDE - Gerência de Decisões
GPP - Gerência de Portfólio de Projetos
GPR - Gerência de Projetos
GQA - Garantia da Qualidade
GRE - Gerência de Requisitos
GRH - Gerência de Recursos Humanos
GRI - Gerência de Riscos
GRU - Gerência de Reutilização
IBGE – Instituto Brasileiro de Geografia e Estatística
IBM - *International Business Machines Corporation*
IEC - *International Electrotechnical Commission*
ISO – *International Organization for Standardization*
ITP - Integração do Produto
MA - Método de Avaliação
MED - Medição
MPS – Melhoria do Processo de Software
MPS.BR - Melhoria do Processo de Software Brasileiro
MR-MPS-SW - Modelo de Referência MPS para Software
PCP - Projeto e Construção do Produto
PDCA - *Plan, Do, Check, Act* (Planejar, executar, checar e ajustar)
PMBOK - *Project Management Body of Knowledge* (Conhecimento em Gerenciamento de Projetos)

PMI - *Project Management Institute* (Instituto de Gerenciamento de Projetos)

PO - *Product Owner* (Dono do produto)

RAP - Resultado de Atributo de Processo

RH - Recursos Humanos

RUP - *Rational Unified Process*

SEI - *Software Engineering Institute* (Instituto de Engenharia de Software)

SM - *Scrum Master*

SPCMM - *Software Process Capability and Maturity Models* (Modelos de Capacidade e Maturidade de Processos de Software)

SPI - *Software Process Improvement* (Melhoria de Processo de Software)

SPICE - *Software Process Improvement and Capability determination* (Melhoria no Processo de Software e Determinação da Capacidade)

SWOT - *Strengths, Weaknesses, Opportunities and Threats*. (Forças, Fraquezas, Oportunidades e Ameaças).

TI - Tecnologias de Informação

UML- *Unified Modeling Language* (Linguagem de Modelagem Unificada)

VAL - Validação

VER - Verificação

SUMÁRIO

1	INTRODUÇÃO	25
1.1	CONTEXTUALIZAÇÃO	25
1.2	CARACTERIZAÇÃO DO PROBLEMA	27
1.3	OBJETIVOS	28
1.3.1	Objetivo geral	28
1.3.2	Objetivos específicos	28
1.4	JUSTIFICATIVA	28
1.5	MOTIVAÇÃO	29
1.6	ESTRUTURA DAS SEÇÕES	30
2	REVISÃO DA LITERATURA	31
2.1	PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE	31
2.2	MELHORAMENTO DE PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE	37
2.3	MODELOS DE MATURIDADE DE DESENVOLVIMENTO DE SOFTWARE	40
2.3.1	ISO/IEC 12207	41
2.3.2	ISO/IEC 33020: 2015	42
2.3.3	CMMI-DEV	43
2.3.4	O programa MPS.BR	45
2.4	O MODELO DE REFERÊNCIA MPS PARA SOFTWARE (MR-MPS-SW)	49
2.4.1	O nível G do MR-MPS-SW	51
2.4.1.1	gerência de Projetos (GPR)	52
2.4.1.2	gerência de Requisitos (GRE)	54
2.4.2	Práticas e ferramentas utilizadas	56
2.4.2.1	scrum	56
2.4.2.2	guia PMBOK	59
2.4.2.3	RUP	60
2.4.2.4	comparação entre o Guia PMBOK, RUP E Scrum	60
2.4.3	Ferramenta 5W2H	61
2.5	MODELO DE AVALIAÇÃO MPS (MA-MPS)	62
2.6	CONSIDERAÇÕES FINAIS DO CAPÍTULO	65
3	METODOLOGIA	66
3.1	CARACTERIZAÇÃO CIENTÍFICA DO TRABALHO	66
3.2	ETAPAS METODOLÓGICAS DO TRABALHO	66
3.3	MÉTODO DE AVALIAÇÃO UTILIZADO NO ESTUDO DE CASO	68

3.4	DELIMITAÇÕES DO TRABALHO.....	70
4	O ESTUDO DE CASO	71
4.1	O AMBIENTE DO ESTUDO DE CASO.....	71
4.2	O PROCESSO DO PROJETO.....	72
4.3	AVALIAÇÃO DO PROCESSO.....	78
4.3.1	Considerações quanto a Gerenciamento de Projetos.....	78
4.3.2	Considerações quanto a Gerenciamento de Requisitos ...	86
4.3.3	A avaliação quanto ao nível G do MR-MPS-SW	87
4.4	PROPOSTAS DE MELHORIA	90
4.4.1	GPR4	90
4.4.2	GPR5 E GRE2.....	91
4.4.3	GPR6 E GPR15.....	94
4.4.4	GPR 12.....	96
4.4.5	5W2H das propostas de melhoria.....	97
5	CONCLUSÕES.....	100
	REFERÊNCIAS BIBLIOGRÁFICAS.....	103
	GLOSSÁRIO.....	108
	APÊNDICE A - Formulário para entrevistas.....	111
	APÊNDICE B - Tabela de avaliação dos Atributos de Processo ..	114
	ANEXO A - Resumo do Guia de Avaliação.....	116
	ANEXO B - Resumo do Guia de Implantação	121
	ANEXO C - Resumo do Guia de utilização do GitLab.....	129
	ANEXO D - <i>Template</i> do Plano de Trabalho.....	135
	ANEXO E - <i>Template</i> de Sprint de projeto.....	139
	ANEXO F - <i>Template</i> de Ata de Reunião.....	141
	ANEXO G - <i>Template</i> de Ata de reunião de Sprint	142
	ANEXO H - <i>Template</i> de produto no Enterprise Architect.....	143
	ANEXO I - <i>Template</i> de documento de Análise Preliminar de Demanda	144

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Desde a década de 1980, a dinâmica da economia mundial sofreu profundas transformações nos modelos de geração e acumulação de riqueza. Ocorre que, diferentemente do antigo padrão de acumulação baseado em recursos tangíveis e dispersos ao redor do mundo, no atual padrão, o conhecimento e a informação exercem papéis centrais na geração de valor para o mercado. (IBGE, 2009).

As Tecnologias de Informação (TI) são elementos propulsores desse novo padrão, fornecendo sistemas informatizados para as instituições, gerando conforto, praticidade e eficiência na execução das tarefas. Uma infinidade de áreas que compõem a sociedade depende e faz uso intensivo desses softwares, como a maioria das prestadoras de serviços, distribuições industriais, produtos elétricos, sistemas financeiros, indústria de música e, até mesmo, jogos de computador. (SOMMERVILLE, 2011).

Segundo Softex (2016), dentro desse cenário, como qualquer outro mercado de produtos e serviços, para garantir a satisfação dos clientes, as empresas fornecedoras de sistemas de informação encontram alguns desafios cotidianos relacionados à qualidade. O fato é que, o diferencial entre o sucesso e a extinção de uma empresa reside, muitas vezes, na qualidade de seu produto ou serviço e na capacidade de entrega de forma competitiva. (SOFTEX, 2016).

Para alcançar o sucesso, a qualidade em software pode ser alcançada de duas formas: investindo na qualidade do produto, que busca definir e alcançar parâmetros tangíveis do produto; e investindo em qualidade do processo de produção, que busca a melhoria das etapas geradores dos produtos para alcançar melhora na qualidade no produto final. (HUMPHREY, 1989).

O presente estudo será focado na busca da qualidade através da melhoria do processo de desenvolvimento de software.

Dentre as possíveis formas de buscar avanços na qualidade do processo, a implantação de modelos de referência e de maturidade para melhoramento de processos tem se mostrado um caminho bastante utilizado pelas empresas. Pois, esses modelos, segundo o *Software Engineering Institute* (SEI, 2010) são, em linhas gerais, um conjunto estruturado de boas práticas que descrevem as características de processos eficazes.

No entanto, quando a maioria dos engenheiros de software pensa em avaliar e melhorar os processos de software, imagens de consultores de alto nível, estruturas de processos complexas e auditorias caríssimas vêm à mente. Essas estruturas de processo bem estabelecidas, são consideradas de esforço pesado, onerosas e usadas mais na certificação externa e menos na melhoria interna do processo. (FELIZ, 2012).

Dentre essas opções, destaca-se o modelo CMMI, um modelo de qualidade de processo de software que tem como principal meta ajudar a integrar funções organizacionais tradicionalmente separadas, definir objetivos e prioridades na melhoria de processos, prover orientação para processos de qualidade e prover um ponto de referência para avaliação desses processos. (SEI, 2010).

Esse modelo tem a intenção de ser um *framework* de aprimoramento de processos com aplicabilidade ampla por meio de uma variedade de empresas. (AHERN et al., 2003).

No entanto, de acordo com Feliz (2012), por ser muito complexo, ainda assim é considerado de difícil implantação para as empresas com menos recursos. Ocorre que, tanto a avaliação quanto a melhoria do processo de software não precisam ser difíceis. Nessa linha, acrescenta ainda o autor supracitado, as conceituadas organizações de software tornam a melhoria de processos parte integrante de seu desenvolvimento organizacional e cultural.

Baseando-se no CMMI, o MPS.BR foi uma alternativa a esse modelo criada focada no cenário nacional. Esse é um programa que foi apresentado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) e é de leve abordagem, prático e pode ser implementado com menor esforço (SOFTEX, 2016). O MPS.BR possui um conjunto de modelos de melhoria e avaliação de processo, projetado preferencialmente para essas micro, pequenas e médias empresas, de forma a atender melhor às suas necessidades de negócio. (SOFTEX, 2016).

Seguindo como base o autor supramencionado, o número de níveis do MPS.BR é maior que o CMMI o torna de mais fácil implementação e evolução entre os níveis, a um baixo custo. Nessa senda, as avaliações do MPS.BR colaboraram cada vez mais na promoção de adoção de boas práticas de engenharia de software, serviços e as práticas de gestão de Recursos Humanos (RH) na indústria brasileira. Esse modelo representou 69% (sessenta e nove por cento) de todas as avaliações aplicadas ao setor de TI no Brasil em 2018. (SOFTEX, 2019).

Essa pesquisa tem como tema a utilização do nível inicial do Modelo de Referência MPS.BR para Software (MR-MPS-SW), o nível

G, como método de avaliação e aprimoramento no processo de desenvolvimento de um sistema repositório de conteúdo para a educação de trânsito.

O que se pretende ao final do trabalho é identificar quais são os melhoramentos recomendados para esse processo, a partir dos pontos fracos encontrados utilizando o nível G do MR-MPS-SW.

1.2 CARACTERIZAÇÃO DO PROBLEMA

Nos últimos 40 anos, consideráveis pesquisas foram conduzidas para as práticas que geram o maior retorno sobre o investimento em relação à eficiência na remoção de defeitos e à produtividade do desenvolvedor. Para a maioria das organizações de software, o valor e o benefício de avaliar a maturidade organizacional é identificar iniciativas tangíveis de aprimoramento de processos com base nas melhores práticas de engenharia de software bem estabelecidas. (FELIZ, 2012).

Dada a relevância em aproveitar as os pontos fortes do MPS.BR para realizar esse aprimoramento no estudo de caso, foi encontrado um projeto para utilizar como piloto, de uma empresa responsável pelas elaborações de sistemas de informação relacionados ao setor transporte e logística. Por motivos internos, a empresa estudada solicitou anonimato neste trabalho, sendo autorizada apenas a exposição de características relevantes ao estudo.

A empresa, atualmente, tem passado por um crescimento rápido e quase metade, do total de aproximadamente 80 (oitenta) colaboradores, são funcionários novos vinculados a menos de um ano. Especificamente no projeto estudado, apenas dois dos oito colaboradores que iniciaram no projeto estão a mais de um ano na empresa.

A fim de evitar dispersão da equipe, relacionamento instáveis, processos mal estabelecidos e baixo desempenho, constatou-se a necessidade de orientar seus colaboradores para melhoria de produtos e processos. A empresa pretende melhorar suas características de processo: estruturação, monitoramento, melhoramento de forma contínua e produção com eficiência.

A justificativa prática desse estudo, portanto, é a necessidade de identificar a situação atual, ressaltar os problemas e determinar os possíveis pontos de melhorias e boas práticas que orientem ações futuras.

Com esse trabalho, pretende-se responder à pergunta: qual a situação atual do projeto e quais são os melhoramentos recomendados no processo de desenvolvimento utilizado para o software repositório de conteúdo para educação de trânsito?

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Avaliar o processo de desenvolvimento de um sistema repositório de conteúdo para apoiar a educação de trânsito com base no nível G de maturidade do MR-MPS-SW, com análise dos resultados e propostas de melhorias no processo de construção de software.

1.3.2 Objetivos específicos

- A. Investigar as formas de melhorias dos processos de software utilizadas atualmente pelas empresas do país;
- B. Mapear os processos necessários para a implantação do MPS.BR nível G como referência;
- C. Levantar a situação atual do processo de produção de software do projeto;
- D. Verificar a conformidade desse processo com o nível G do modelo de desenvolvimento de software do MPS-BR;
- E. Propor melhorias de procedimentos que agregam valor aos produtos desenvolvidos com foco nos clientes;

1.4 JUSTIFICATIVA

A importância de melhorar software utilizando o MR-MPS-SW é relativa ao diferencial que ele oferece, equiparado aos outros modelos. O MPS.BR é um projeto de fundamentação técnica mais abrangente, indo além da referência em que foi embasado, o CMMI. (LIMA, 2017). É também compatível com abordagens relevantes do cenário mundial para determinação, avaliação e aperfeiçoamento de processos de software, como ISO/IEC 12207 e ISO/IEC 33020, e é apropriado às empresas com diversos portes e particularidades. (SOFTEX, 2019).

Ainda, Lima (2017) ressalta que seus componentes se segmentam em guias de referência, implementação e avaliação, tornando seu alcance abrangente e flexível. Por ter farta referenciação, analisa e direciona as empresas desde processos e serviços até qualidade dos produtos.

O MPS.BR, também, possui uma bagagem de benefícios de experiências passadas de uma comunidade, uma linguagem compartilhada com os outros modelos, um método de trabalho para priorização das

atividades e uma forma própria de balizar melhoria para uma organização. (SEI, 2010).

Os efeitos de implantar o MPS.BR são proveitosos à busca por projetos que satisfazem seus clientes. Isso porque as vantagens também são percebidas a curto e longo prazo. Dentre os prós pode-se enumerar os principais, que são também os mais buscados pelas empresas: aumento de produtividade, na satisfação dos clientes, na definição de prazos e no faturamento. (FUMSOFT, 2011).

O bom custo/benefício também é claro. É um método mais barato de avaliação e bastante semelhante às suas alternativas. A maturação que a organização pode obter com sua utilização e com avanço nos níveis do modelo resulta em sistemas de informação com padrões de qualidade específicos, importantes e que utilizam boas práticas de desenvolvimento de software. Esses são fatores determinantes para aumento de eficiência. (LIMA, 2017).

Tendo em vista os aspectos apresentados, uma das justificativas deste trabalho é a diversidade de vantagens que se apresentam da utilização do MR-MPS-SW em uma empresa de tecnologia de informação.

No entanto, mesmo sendo um dos métodos mais usados e estabelecidos, os números indicam que há uma baixa aderência a esse modelo. O estudo realizado pela Associação Brasileira das Empresas de Software (ABES) apontou que, até 2018, haviam mais de 19 mil empresas dedicadas ao desenvolvimento, produção, distribuição de software e de prestação de serviços no mercado nacional. E, o estudo mostra que esse número só cresce.

Existem, contudo, vigentes para 2019, apenas 100 avaliações publicadas no MPS-SW: 36 do nível G; 34 do nível F; 3 do nível E; 0 do nível D; 23 do nível C; 0 do nível B; e 2 do nível A. Dessas, 39% são do Sul do Brasil. (SOFTEX, 2019).

Diante disso, pode-se observar que há um número baixo de organizações desenvolvedoras de software que comprovadamente buscam qualidade do processo no Brasil através do MPS.BR. Das poucas que procuram se certificar, a maioria (85%) encontra-se nos dois níveis mais baixos, G e F. (SOFTEX, 2019).

As evidências de pouca prática dos projetos de software no Brasil com o modelo e práticas existentes relacionados a ele, portanto, são também fortes justificativas para este trabalho.

1.5 MOTIVAÇÃO

A melhoria de software não é um assunto novo para a maioria dos profissionais de tecnologia da informação, porém, carece de patrocínio e de iniciativas para a sua execução e manutenção. Tomei conhecimento do MPS.BR por meio de estágio e é de meu interesse acadêmico me aprofundar nesse assunto.

Comparar o processo de desenvolvimento de software da instituição estudada ao modelo de maturidade MPS.BR e fornecer recomendações, possibilitará a reavaliação dos seus processos atuais. Se aprovadas, será possível reforçar uma atividade ainda não institucionalizada: a melhoria contínua.

Para mim, investir em aperfeiçoamento, tanto profissional quanto pessoal, é visto como um meio para outros fins. Ademais, a busca pela excelência é por si só é um objetivo de grande valor.

Essas são, com certeza, as maiores fontes motivacionais deste trabalho.

1.6 ESTRUTURA DAS SEÇÕES

O Capítulo 1 abordou, de forma introdutória, uma visão geral a respeito do trabalho desenvolvido neste trabalho. São identificados: o contexto em que o caso está inserido, a caracterização do problema para o estudo de caso, os objetivos definidos e as justificativas relacionadas a escolha do problema.

No Capítulo 2, realizou-se uma revisão da literatura, onde será apresentada a fundamentação teórica relacionada com o tema de melhoria de desenvolvimento de software. Serão ainda abordados conceitos relacionados a processos de desenvolvimento de software sob o nível G de MR-MPS-SW.

No Capítulo 3, apresentou-se o modelo metodológico de implantado neste trabalho, o MR-MPS-SW, bem como os procedimentos de avaliação do processo de produção de software do projeto, de implantação do método e de acompanhamento.

No Capítulo 4, apresentou-se o processo atual utilizado no projeto. Discorrer-se-á sobre a avaliação realizada sob a análise da metodologia desenvolvida e os resultados obtidos. Por fim, serão apresentadas propostas de melhorias para os pontos fracos identificados na avaliação.

Ao final, no Capítulo 5 apresentou-se conclusões sobre os resultados obtidos na implantação, seguido de um resumo da pesquisa, contribuições, limitações, possibilidades de trabalhos futuros e considerações finais sobre o trabalho.

2 REVISÃO DA LITERATURA

Neste capítulo realizou-se uma revisão da literatura no tocante aos conceitos apresentados relacionados aos processos produtivos de desenvolvimento de software, melhoramento de processos de desenvolvimento de software e avaliação desses processos.

Ainda nesse Capítulo, será então apresentado o programa MPS.BR, sua base técnica e seu modelo de referência para desenvolvimento de softwares, o MR-MPS-SW. Ademais, será então aprofundado sobre o nível inicial do modelo, o nível G do MPS.BR, utilizado no estudo de caso.

2.1 PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

Por muitos anos, sobretudo nas décadas de 1980 e 1990, desenvolvia-se sistema operacional executando um planejamento minucioso no início do projeto, como é feito na indústria de construção civil na atualidade. Nesse tempo, acreditava-se que a maneira mais correta para conseguir o melhor software era utilizar, durante o estudo desses projetos, métodos minuciosos e controlados de concepção e processos rigorosos de desenvolvimento. (SOMMERVILLE, 2011).

Tal pensamento, seguindo o entendimento do autor citado acima, era adequado ao desenvolvimento de sistemas críticos, tais como aeroespacial ou de segurança nacional dos Estados Unidos, e que poderiam levar diversos anos desde sua concepção inicial até a implantação final. A partir desse nicho que foi difundida essa abordagem tradicional de desenvolvimento de software para o resto da comunidade de engenharia de software. (SOMMERVILLE, 2011).

Nas metodologias tradicionais, as diferentes etapas do processo são bem definidas e fortemente prescritivas. Conforme Sabbagh (2013), cada etapa do desenvolvimento está fortemente ligada à etapa anterior, ao ponto de os resultados obtidos em uma etapa fornecem as informações para planejamento da etapa seguinte.

As abordagens tradicionais podem ser divididas em dois modelos: linear e incremental. O modelo linear trabalha com uma entrega única ao cliente, ao final do projeto. O método incremental, trabalha com entregas parciais, chamadas de incrementos. (WYSOCKI, 2014). As diferenças nas etapas desses dois métodos podem ser visualizadas na Figura 1.

No entanto, esse tipo de tratamento gera um sobrepeso de responsabilidade para o planejamento do projeto e para a documentação

inicial. Ainda que justificável para projetos críticos, bastante previsíveis ou com muitos envolvidos no ciclo de vida do produto, não são eficientes para ambientes com mudanças constantes e com imprevisibilidade. (SABBAGH, 2013).

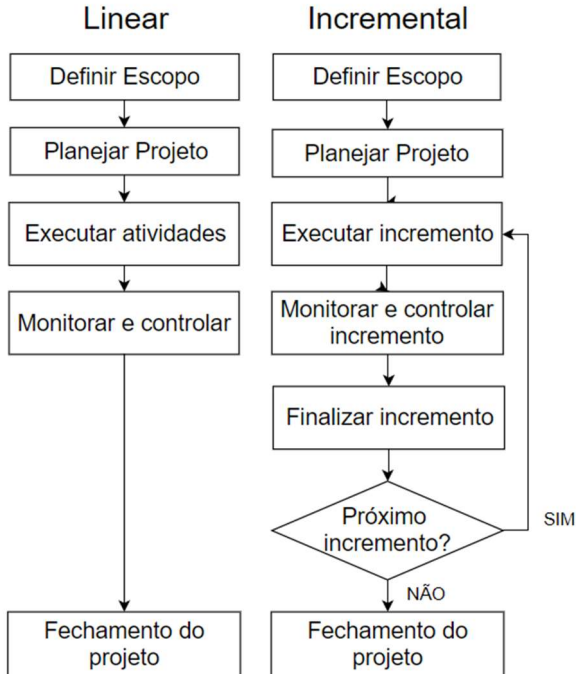


Figura 1 - Ciclos de vida tradicionais
Fonte: adaptado de Wysocki (2014).

Na visão de Sommerville (2011), esse é o caso do maior número dos projetos de desenvolvimento de software, onde é comum que os requisitos sejam alterados no decorrer do desenvolvimento e tragam retrabalho de planejamento e documentação. Nesses projetos, é requerido um tratamento empírico e mais adaptativo.

Além disso, o emprego da abordagem tradicional não é justificado principalmente para projetos de menor porte, pois resulta em gastos de mais tempo na burocrática análise dos requisitos do que no próprio desenvolvimento do software. (SOMMERVILLE, 2011).

Todos esses fatores ocasionaram um grande descontentamento e na descoberta de alternativas mais adequadas para esses casos: os chamados métodos ágeis. As metodologias ágeis foram desenvolvidas

para sanar estas fraquezas encontradas na abordagem tradicional. (PRESSMAN, 2011). Conforme o entendimento de Siqueira (2004), essas metodologias se inseriram melhor ao ambiente turbulento dos negócios atuais, que com frequência exigem mudanças rápidas. Para tal, as metodologias ágeis utilizam de princípios e práticas bem divergentes dos métodos tradicionais. (SIQUEIRA, 2004).

Em 2001, segundo Pressman (2011), ocorreu um dos principais marcos do desenvolvimento do software: a criação do "Manifesto Ágil". Esse, estava associado a um movimento de dezessete renomados autores, desenvolvedores e consultores, que se juntaram para discutir os métodos utilizados na resolução de problemas no tocante a abordagem tradicional, cujo objetivo era atualizar o desenvolvimento de software no cenário dos negócios. (PRESSMAN, 2011).

Nesse manifesto estão contidos os valores e princípios que norteiam os métodos ágeis. É destacado principalmente que, para o sucesso do projeto de desenvolvimento, são mais importantes os indivíduos e interações eles, software em funcionamento, colaboração com o cliente e responder às mudanças. O valor de processos, ferramentas, documentação, contratos e planos, apesar de ainda serem essenciais, são menos valiosos para o êxito do projeto. (BECK, BEEDLE, et al., 2001).

Wysocki (2014) separa a abordagem ágil em dois métodos: iterativos e adaptativos. Esses métodos não se diferenciam muito em suas etapas, mas no entendimento da solução final. No método iterativo a organização tem uma compreensão um pouco melhor da solução que será desenvolvida. Já o método adaptativo se diferencia por possuir uma menor compreensão dessa solução.

Para casos em que não há entendimento claro nem mesmo do escopo, a metodologia adequada é chamada de extrema. Nela, até mesmo o escopo é redefinido em cada iteração. (WYSOCKI, 2014).

A Figura 2 ilustra as etapas dos métodos ágeis e extremos.

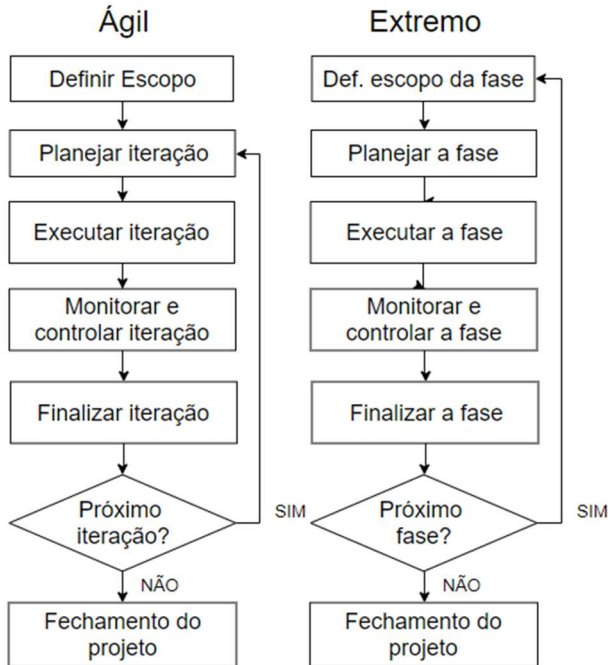


Figura 2 - Ciclos de vida ágeis e extremos.

Fonte: adaptado de Wysocki (2014)

Compreendidos os ciclos de vida dos projetos, é importante compreender os conceitos específicos relacionados ao processo de desenvolvimento de software.

De forma sucinta, o processo de software trata-se de um padrão que determina tarefas, regras, ações e restrições que, caso seja seguido de forma adequada, irá resultar em um produto de software, considerando todos os requisitos do cliente e seus objetivos. (HUMPHREY, 1989). Esse mesmo autor é enfático que é essencial o foco nas necessidades do cliente.

Já para o *Software Engineering Institute* (SEI, 2010), um processo é composto pelo conjunto do que é feito, quando, por quem, os itens utilizados e os itens produzidos.

De acordo com Pressman (2011), o processo de software, em geral, é um *framework* de atividades, tarefas e ações necessárias para a construção de software de alta qualidade.

Falbo (1998) e Sommerville (2011) apresentam definições importantes para a entendimento do processo de desenvolvimento software:

1. Atividades: trabalhos ou tarefas prescritas para realização. Utilizam recursos e podem requerer ou gerar um ou mais artefatos.
2. Artefatos: produtos de software ou documentações em geral geradas por atividades ou consumidas por elas ao longo de sua realização;
3. Procedimentos: são condutas padronizadas e bem estabelecidas para ordenar a execução de tarefas. Ferramentas de software podem apoiar a adesão aos procedimentos;
4. Recursos: pessoas, ferramentas, equipamentos, etc. São insumos necessários à execução de uma atividade;
5. Processos: conjunto de atividades correlacionadas orientadas ao desenvolvimento de um produto;
6. Papéis: indicam as responsabilidades das pessoas envolvidas no processo;
7. Pré e pós-condições: podem ser definidas antes e depois de uma atividade do processo ou da produção de um produto.

Além disso, Pressman (2011) acredita ser possível subdividir esse processo em cinco etapas básicas:

1. Comunicação: em que é estabelecida uma comunicação com os *stakeholders*, principalmente o cliente, para entender o escopo, seus objetivos, necessidades e seus requisitos para os projetos;
2. Planejamento: quando são definidas as tarefas que serão conduzidas, estudados os riscos, balanceamento dos recursos necessários e um cronograma geral de trabalho;
3. Modelagem: etapa que é concebido um protótipo, esboçadas suas funcionalidades, de forma a facilitar o entendimento do produto final.
4. Construção: quando o planejado é executado e testado, para efetivamente construir o produto;
5. Entrega: o que foi criado será entregue ao cliente, avaliado e recebido um *feedback*.

Para Pressman (2011), cada etapa contém um grupo de operações e tarefas que devem ser concluídas, artefatos que serão gerados e aspectos

relevantes para garantir a qualidade e cumprimento dos marcos usados para verificar o progresso do projeto. Destaca, também, a relevância de um fluxo de processo, para definir a relação entre as atividades.

Esse mesmo autor destaca esse fluxo deve definir a ordem em que essas atividades serão executadas e as condições necessárias para o avanço na execução. Afirma ainda que em alguns casos, comuns nas metodologias tradicionais, a execução das atividades é feita de modo sequenciado, mas nos métodos ágeis é mais comum um fluxo iterativo, em que há repetição de ciclos de atividades, chamadas iterações, de acordo com critérios definidas previamente no processo. Não existe, portanto, um processo ideal para todas as organizações e projetos, sendo crucial que cada organização acabe criando os seus próprios. (PRESSMAN, 2011).

Sommerville (2011) elucida que, determinar um processo de software e o fluxo que será operado são decisões complexas que demandam avaliar vários elementos de uma organização. Para corroborar com essa afirmação, traz como exemplos desses elementos as pessoas envolvidas, as particularidades do produto, os recursos a disposição para a implementação do processo e os projetos que serão processados. Por tal motivo, esclarece que muitas vezes as empresas precisam estipular alternativos processos e fluxos para diferentes projetos.

Os sistemas críticos são exemplos em que se percebe a necessidade de utilizar um processo que seja muito bem sistematizado, com passos bem definidos e inflexíveis, ou seja, métodos tradicionais. (SOMMERVILLE, 2011).

Sommerville (2011) também explica, que para softwares de negócios, em que o comprador se mostra indeciso e pode ter modificações imprevisíveis nos requisitos, existe a necessidade de se usar um processo mais flexível, ou seja, métodos ágeis. Em casos ainda mais incertos, deve-se aplicar o método extremo.

Corroborando com esse entendimento, Falbo (2005) afirma que, para ser eficaz, não importa se for com aspecto tradicional, ágil ou extremo, um processo de software deve se adequar às especificidades do projeto, dependendo do caso. (FALBO, 2005). Ademais, desenvolvimento de software são processos complexos, e precisam que decisões e julgamentos sejam tomados constantemente (SOMMERVILLE, 2011).

Ainda assim, é possível buscar alguns padrões de processo em etapas específicas dentro de uma organização. Humphrey (1989) destaca os principais motivos para padronizar um processo, quando possível:

1. reduz problemas relacionados a treinamento, revisão e utilização de ferramentas;
2. aproveita *know how* adquirido sobre processos específicos, melhorando o processo geral;
3. proporciona uma base para medições de processo e qualidade; e,
4. reduz a impraticabilidade de criar do zero um novo processo para cada projeto que irá ser executado.

2.2 MELHORAMENTO DE PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

Organizações que desenvolvem software encaram os mesmos desafios que qualquer outro segmento do mercado, que são: o melhorar a qualidade do produto, a diminuir os custos e o atender o prazo fixado (PRESSMAN, 2011).

Face a isso, Pressman (2011) traz que a busca por um software de qualidade tem seguido duas vertentes: a qualidade do processo e a qualidade do produto. A qualidade do produto foca em determinar características tangíveis dos softwares desenvolvidos, criando critérios que apoiam na avaliação do que foi produzido. Por outro lado, a qualidade do processo foca nas tarefas geradoras desse produto, utilizando de controle e gerenciamento do ciclo de vida do software. (PRESSMAN, 2011).

Humphrey (1989) acrescenta que a qualidade do processo de software é a abordagem popularmente utilizada. Isso porque a relação entre atividades e componentes podem exibir grande complexidade na disposição de tarefas durante o ciclo de vida do software. Assim, alcançar um bom processo de software possibilita que as equipes da engenharia de Software atuem de forma mais organizada. (HUMPHREY, 1989).

Acrescendo a isso, Cavalcanti (2012) comenta que os processos devem ser tecnicamente corretos e capazes de atender às necessidades do negócio. Entretanto, assim entende o autor, os processos podem estar adequados do ponto de vista da Engenharia de Software e não serem eficientes, podendo consumir muito tempo e esforço ou não produzirem produtos com o nível de qualidade adequada à satisfação das necessidades dos usuários.

Para mitigar isso, devem também, entre seus esforços de definição de processo, o de melhoria contínua. É importante, neste contexto, dispor-se de mecanismos capazes de evidenciar problemas nos processos e apoiar na identificação de pontos de melhoria (CAVALCANTI et al., 2012). Atentas a isso, verifica-se, nos últimos anos, um aumento nas

iniciativas das organizações para a adoção de práticas de melhoria de processos de software. (PRESSMAN, 2011).

Sob a perspectiva do autor supracitado, em alguns casos estas práticas são feitas de formas pouco controladas, não passíveis de serem reproduzidas e ligados ao talento individuais dos membros da equipe. Outros casos, por iniciativas improvisadas que por acaso alcançam o resultado desejado. No entanto, as que são mais efetivas são aquelas que se baseiam em práticas de engenharia de software já consolidadas no mercado. (PRESSMAN, 2011).

Sendo assim, Sommerville (2011) define melhoria de processo de software, conhecida também como SPI (*Software Process Improvement*), como uma forma de compreender os processos existentes na organização e modificá-los para obter melhores resultados na qualidade dos produtos e redução na quantidade de defeitos no software entregue, que resulta em redução dos custos e tempo de desenvolvimento. Assim, a diminuição dos defeitos processo, até que possa ser replicado com precisão, torna o resultado esperado do processo algo previsível e padronizado. (SOMMERVILLE, 2007).

Pode-se ratificar isso com a afirmação de Pressman (2011), que salienta como a aplicação de melhoria de processo de desenvolvimento de software transforma a abordagem existente em algo mais focado, mais confiável, em relação à qualidade de produto e prazo de entrega, e passível de uma melhor repetitividade em projetos de características semelhantes. Para ele, partindo-se da premissa que os elementos de um processo foram bem definidos, a abordagem organizacional para o desenvolvimento pode ser avaliada em relação aos seus elementos e uma estratégia significativa para a melhoria pode ser tomada.

Para tanto, colhe-se do livro do autor supramencionado a Figura 3 que apresenta uma visão geral da uma estrutura SPI típica, cujo objetivo principal é a definição do processo de software.

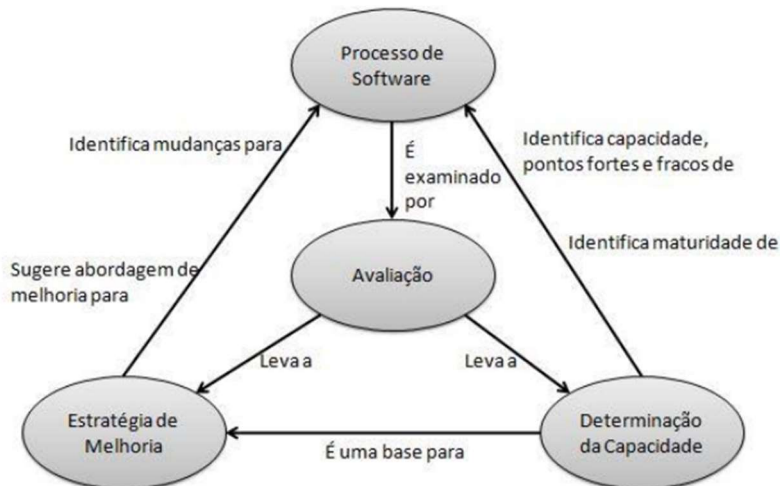


Figura 3 - Estrutura de melhoria do processo de software.

Fonte: Pressman (2011)

Dentro desse prisma, Pressman (2011) pontua que a partir do processo inicial é realizada uma avaliação para determinação da capacidade de processo, pontos fortes e fracos. Essa avaliação conduz à um Estratégia de melhoria, que sugere mudanças e apresenta abordagens de melhoria para o processo.

Destaca ainda que sempre há espaço para melhorias no processo existente, que é possível alcançar uma versão mais refinada, otimizada e ajustada à realidade da organização. (PRESSMAN, 2011).

Apoiando essa ideia, pode-se citar Alves (2008), que discorre que a partir do momento que a organização entende que processos de fabricação de software inadequados afetam a imagem da empresa e elevam os custos de desenvolvimento, ela deve implantar um processo de melhoria contínua para o processo de desenvolvimento software.

Segundo Sommerville (2011), essa melhoria de processo de software é composta por um conjunto cíclico de atividades e deve conter meios de monitorar a qualidade do processo embasando sua análise na mensuração do número de problemas do produto.

Ademais, segundo Alves (2008), atividades de prevenção de defeitos e riscos são mais eficientes que as de correção. Para ele, uma hora voltada a prevenção de defeitos do sistema nas etapas iniciais do ciclo de desenvolvimento diminui de 3 a 10 horas na correção posteriores a execução de tarefa errada. Também, declara que 40% a 50% do custo

total dos projetos de software são retificando defeitos no software. Dessa forma, é possível verificar que investir na garantia da qualidade compensa rapidamente. (ALVES, 2008).

Portanto, de acordo com Pressman (2011), a melhoria de processo de software aumenta o custo dos projetos e garante o retorno do investimento. Ele cita que a propagação de defeitos, qualidade do produto, manutenção e suporte do software e atraso nas entregas são as causas mais notórias na interferência no custo do projeto. Para Pressman (2011), mitigar antecipadamente suas causas proporciona melhora nos indicadores destes quatro fatores.

2.3 MODELOS DE MATURIDADE DE DESENVOLVIMENTO DE SOFTWARE

Modelos de Maturidade e Capacidade de Processos de Software, do inglês *Software Process Capability and Maturity Models* (SPCMM), consistem em um agrupamento de melhores práticas de processos, firmados com base nas experiências prévias e princípios de gestão de processos, colocados sob os conceitos de capacidade e maturidade, adequados para avaliar ou melhorar os processos de uma organização. (HAUCK et al, 2011).

Esses modelos são úteis para medir a maturidade dos processos, avaliar o estado atual das práticas, orientar a evolução da mudança organizacional e possibilitar estabelecer comparações ou *benchmarking*. (MENAYO et al, 2006).

Na indústria de desenvolvimento de software, os modelos de maturidade e capacidade são destinados a garantir que sejam robustos os processos de desenvolvimento e as atividades importantes à sua gestão. Acredita-se que elas podem elevar a performance em relação ao custo, prazo, qualidade, produtividade, satisfação do cliente, retorno do investimento e aumento da vantagem competitiva. (KALINOWSKI et al., 2010).

Um modelo de maturidade de processo de software descreve um processo de software simplificado e é apresentado a partir de uma perspectiva específica. Com o aparecimento de diversos padrões internacionais de maturidade, chamados de modelos de referência, as organizações começaram a utilizar os referidos modelos para redefinir seus processos de software. (SCHEID et al, 2007).

Gonçalo et al (2010) discorrem sobre o fato de que quanto maior o nível de maturidade de uma organização, mais avançada é sua orientação estratégica e sua capacidade de aplicação dos processos

relacionados, o que possivelmente, implica melhores resultados.

Nesses modelos, portanto, a qualidade do produto é reflexo da maturidade do processo de software de uma organização as estruturas e os parâmetros dos modelos são formalizados e delineados pelos grupos que as desenvolvem. (PRESSMAN, 2011).

Pressman (2011) afirma que quanto mais evoluído o processo, mais maduro ele é julgado. Logo, segundo Albuquerque (2008), ao buscar a melhoria de processos de software utilizando essas abordagens, deve-se procurar avançar aos níveis mais altos de maturidade.

Parafraseando Pressman (2011), um modelo de maturidade é encaixado no contexto de uma estrutura de melhoria de processo de software com o propósito de prover um parâmetro de comparação para indicar de forma geral o nível de maturidade de um processo estudado.

Para alcançar esse propósito, os modelos contêm elementos fundamentais para processos efetivos, para uma ou mais disciplinas. Descrevem uma trajetória de melhoria evolutiva, geralmente separada em níveis ou escala ordinal, desde processos imaturos até processos maduros e disciplinados. (SEI, 2010).

É importante ressaltar que as boas práticas descritas por estes materiais de apoio à melhoria do processo não têm o intuito de definir o processo na organização, pois este deve-se adequar à realidade de cada organização. (SPINOLA et al., 2008).

Os esforços no sentido de auxiliar na melhoria de qualidade dos processos de software, através de práticas de engenharia de software, têm sido propostos desde a década de 90 através de modelos de maturidade internacionais. (ALVES, 2008).

Atualmente, estão disponíveis no cenário internacional uma grande diversidade de modelos, como o SPICE (*Software Process Improvement and Capability Determination*) e o CMMI (*Capability Maturity Model Integration*). Mais recentemente surgiu, no Brasil, o MPS.BR (Melhoria do Processo de Software Brasileiro). (PRESSMAN, 2011).

A seguir, nos subitens da seção 2.3, serão apresentados alguns desses modelos e normas, relevantes para esse trabalho. São eles: ISO/IEC 12207, ISO/IEC 33020, CMMI-DEV e, finalmente, o MPS.BR.

2.3.1 ISO/IEC 12207

A Norma Internacional ISO/IEC 12207 foi publicada pela ISO – *International Organization for Standardization* (SOFTEX, 2016). Sua primeira versão foi lançada em 1995, atualizada em 2008 e, atualmente,

está em vigor em uma versão revisada em 2017. (ISO/IEC, 2017). A versão apresentada neste trabalho é a de 2008, com parceria com e o IEC - *International Electrotechnical Commission*, já que foi a versão baseada para criação do MPS.BR. (SOFTEX, 2016).

Logo, a ISO / IEC 12207 (2008, pag. 3) tem o objetivo de estabelecer:

“[...] uma estrutura comum para processos de ciclo de vida de software, com terminologia bem definida, que pode ser referenciada pela indústria de software. A estrutura contém processos, atividades e tarefas que servem para ser aplicadas durante a aquisição de um produto ou serviço de software e durante o fornecimento, desenvolvimento, operação, manutenção e descarte de produtos de software.”

Consoante, colhe-se da Softex (2016) explicações de que a norma foi construída para que fosse utilizada em todo o ciclo de vida de software, ou seja, desde o início (concepção) até a descontinuidade do produto de software; e por todos os envolvidos com a produção, manutenção e operação do software que são: adquirentes, fornecedores, operadores, desenvolvedores, mantenedores, gerentes, profissionais de qualidade e usuários.

2.3.2 ISO/IEC 33020: 2015

A Norma Internacional ISO/IEC 33020 (ISO/IEC, 2015) foi publicada em 2015 e substitui e amplia algumas partes da ISO/IEC 15504, que por sua vez originou-se do projeto SPICE, e estabelece uma estrutura de medição de processo que suporta a avaliação da capacidade do processo. (SOFTEX, 2016).

De acordo com a ISO/IEC (2015), as estruturas de medição de processo definidas na ISO / IEC 33020: 2015: facilitam a autoavaliação, fornecem uma base para o uso na melhoria do processo e determinação da qualidade do processo, são aplicáveis a todos os domínios e tamanhos de organização da aplicação, produzem um conjunto de classificações de atributos de processo (capacidade, perfil de processo) e derivam um nível de capacidade do processo.

De acordo com (SOFTEX, 2016, pag. 14):

“O conjunto de normas que fornecem um framework consciente e coerente para avaliação de características de qualidade de processos, fornecendo um conjunto claro de requisitos para o processo de avaliação e os recursos necessários para implementá-lo efetivamente é conhecido como "família 330xx", o qual resta-se à realização de avaliações de processos de software com dois objetivos: a melhoria de processos e a determinação da capacidade de processos de uma unidade organizacional.”

Se o objetivo for a melhoria de processos, a unidade organizacional pode realizar uma avaliação com o objetivo de gerar um perfil dos processos, com pontos fortes e os pontos fracos, que será usado para a elaboração de um plano de melhorias. No segundo caso, o perfil de capacidade permite ao contratante estimar o risco associado à contratação daquele fornecedor em potencial para auxiliar na tomada de decisão de contratá-lo ou não. (SOFTEX, 2016)

Resumidamente, o modelo é composto por 5 processos, Cliente-Fornecedor, Engenharia, Suporte, Gerência e Organização, contendo 6 níveis de capacidade para cada processo: incompleto, executado, gerenciado, estabelecido, previsível e otimização (SOFTEX, 2019).

2.3.3 CMMI-DEV

A universidade Carnegie-Mellon, nos Estados Unidos, criou o *Capability Maturity Model Integration* (CMMI) visando oferecer um conjunto comprovado de melhores práticas globais que impulsiona o desempenho dos negócios através da criação e comparação de recursos-chave. Foi originalmente criado para o Departamento de Defesa dos EUA para avaliar a qualidade e a capacidade de seus fornecedores de software. (CMMI INSTITUTE, 2019).

O CMMI indica uma série de recursos de engenharia de software que precisam estar presentes em uma organização, ao passo em que ela alcança diferentes níveis de maturidade do processo. (SOMMERVILLE, 2010).

Atualmente, o CMMI oferece três modelos: CMMI *Supplier Management* (CMMI para Gerenciamento de Fornecedores), CMMI *Development* (CMMI para Desenvolvimento, ou, CMMI-DEV) e CMMI *Services* (CMMI para Serviços). O modelo CMMI para Gerenciamento de Fornecedores tem como foco atividades de iniciação e gerência de

aquisição de produtos e serviços em uma organização. Já o modelo CMMI para Serviços foca em atividades para fornecer serviços de qualidade a usuários finais. (CMMI INSTITUTE, 2019).

O foco do CMMI-DEV, o modelo relevante para esse trabalho, se dá nas atividades de desenvolvimento de produtos e serviços de qualidade para que se atinja as necessidades de clientes e usuários finais (CMMI INSTITUTE, 2019). Este trabalho foca nas características da versão do CMMI-DEV de 2010, que foi utilizada como base para o MPS.BR. (SOFTEX, 2016).

Os seus níveis de maturidade seguem cinco níveis de maturidade, partindo do nível 1 até o nível 5, conforme a Figura 4.



Nível de maturidade 1: Inicial

Imprevisível e reativo. O trabalho é concluído, mas geralmente é atrasado e excede o orçamento.



Nível de maturidade 2: Gerenciado

Gerenciado no nível do projeto. Os projetos são planejados, executados, medidos e controlados.



Nível de maturidade 3: Definido

Proativo, em vez de reativo. Os padrões de toda a organização fornecem orientação sobre projetos, programas e portfólios.



Nível de maturidade 4: Gerenciado quantitativamente

Medido e controlado. A organização é orientada por dados com objetivos quantitativos de melhoria de desempenho que são previsíveis e alinhados para atender às necessidades das partes interessadas internas e externas.



Nível de maturidade 5: Otimizando

Estável e flexível. A organização está focada na melhoria contínua e é construída para dinamizar e responder a oportunidades e mudanças. A estabilidade da organização fornece uma plataforma para agilidade e inovação.

Figura 4 - Níveis de maturidade do CMMI.
Fonte: traduzido de CMMI INSTITUTE (2019)

O CMMI-DEV de 2010 possui 22 áreas de processos as quais

consistem em um conjunto de práticas relacionadas a uma determinada área que, quando implementadas, satisfazem a um conjunto de metas consideradas importantes para realizar melhorias significativas naquela área. O CMMI-DEV contém práticas que abrangem Gerência de Projetos, Gerência de Processos, Engenharia de Sistemas, Engenharia de Hardware, Engenharia de Software e outros processos de suporte e manutenção (CMMI INSTITUTE, 2010).

2.3.4 O programa MPS.BR

De acordo com a Softex (2016, pag. 4):

O Programa MPS.BR é um programa mobilizador, de longo prazo, criado em dezembro de 2003, coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), com apoio do Ministério da Ciência, Tecnologia e Inovação (MCTI), Financiadora de Estudos e Projetos (FINEP), Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) e Banco Interamericano de Desenvolvimento (BID/FUMIN).

Seu objetivo é o aumento da competitividade das organizações brasileiras pela melhoria de seus processos. Os modelos MPS foram projetados para que atendam a necessidade de implantar os princípios de Engenharia de Software de forma adequada ao contexto das empresas. (SOFTEX, 2016).

O Programa MPS.BR conta com a participação de representantes de universidades, instituições governamentais, centros de pesquisa e organizações privadas, os quais contribuem com suas visões complementares que agregam valor e qualidade ao Programa. (SOFTEX, 2016).

Seus modelos de qualidade tem o intuito de ser compatível com as principais abordagens internacionais para definição, avaliação e melhoria dos processos de software. (SOFTEX, 2016).

O programa defende a melhoria do processo gradual, através de níveis de maturidade. Por ser voltado para a realidade brasileira, no qual é bastante custoso evoluir diversos processos concomitantemente, o mesmo possui mais níveis de maturidade que o CMMI. (SOFTEX, 2016). O Nível 2 do CMMI, por exemplo, é equivalente aos níveis G e F do

MPS.BR. Uma comparação ilustrativa desses níveis pode ser verificada na Figura 5. (SOFTEX, 2016)

Os níveis são de G ao A, sendo A o nível mais maduro. São cumulativos, então ao passar de nível G para o nível F, por exemplo, os processos do nível de maturidade G devem também ser atendidos. (SOFTEX, 2016).

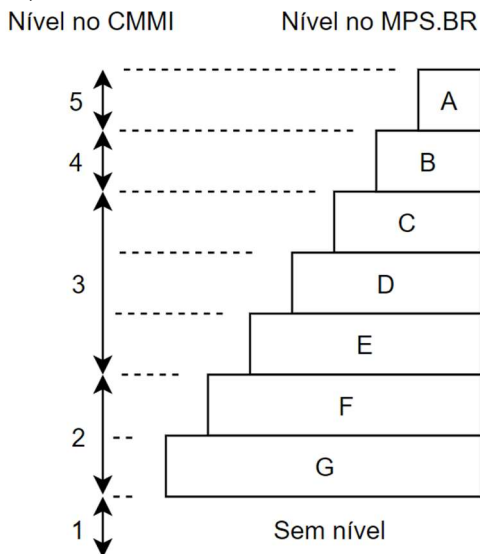


Figura 5 - Comparação dos níveis de maturidade do MPS.BR com do CMMI.

Fonte: elaborado pelo autor baseado em (SOFTEX, 2016).

Os modelos MPS.BR estão descritos por meio de documentos em formato de guias, disponíveis no site da Softex (2019):

- Guia Geral MPS de Software: contém a descrição da estrutura dos modelos MPS e detalha o Modelo de Referência MPS para Software (MR-MPS-SW), seus componentes e as definições comuns necessárias para seu entendimento e aplicação;
- Guia Geral MPS de Serviços: contém a descrição da estrutura dos modelos MPS e detalha o Modelo de Referência MPS para Serviços (MR-MPS-SV), seus componentes e as definições comuns necessárias para seu entendimento e aplicação;
- Guia Geral MPS de Gestão de Pessoas: contém a descrição da estrutura dos modelos MPS e detalha o Modelo de Referência MPS para Gestão de Pessoas (MR-MPS-RH), seus

componentes e as definições comuns necessárias para seu entendimento e aplicação;

- Guia de Avaliação: descreve o processo e o método de avaliação MA-MPS, os requisitos para a constituição de Instituições Avaliadoras e para o credenciamento de avaliadores MA-MPS;
- Guias de Implementação de Software: conjunto guias, separados pelos níveis, que fornecem orientações para implementar o MR-MPS-SW;
- Guias de Implementação de Serviços: conjunto guias, separados pelos níveis, que fornecem orientações para implementar o MR-MPS-SV;
- Guia de Aquisição: descreve um processo de aquisição de software e serviços correlatos.

A Figura 6, organiza esses modelos mencionados pela Softex (2016), bem como suas bases técnicas e seus respectivos guias.

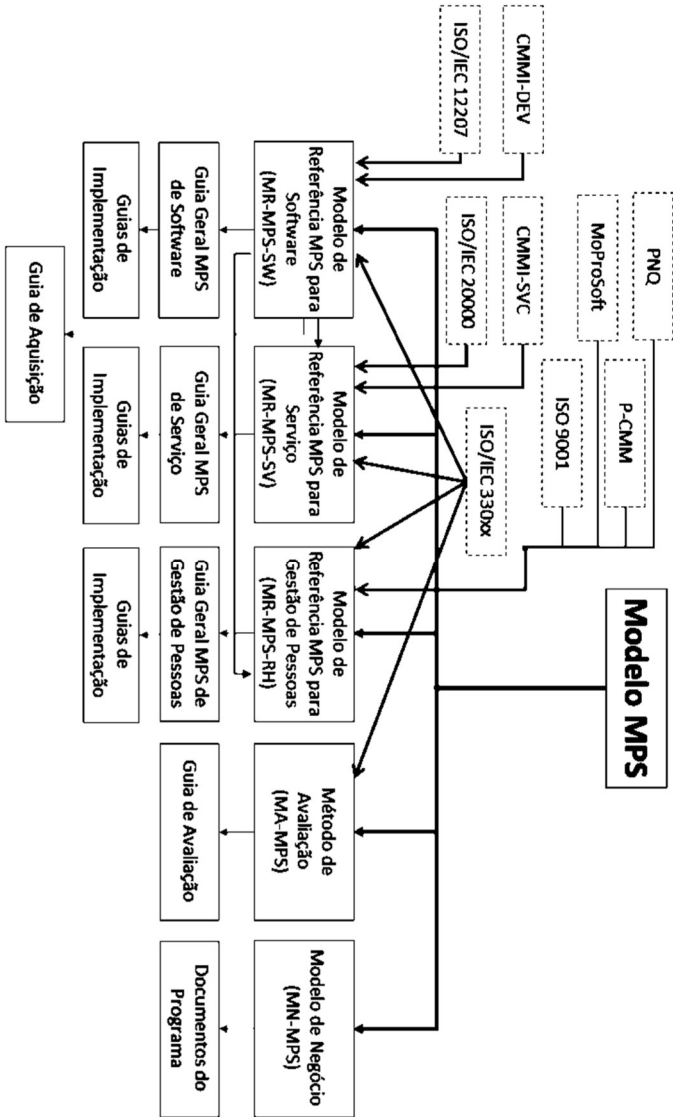


Figura 6 - Componentes do Programa MPS.BR.
 Fonte: Softex (2016).

Esse trabalho trata de um projeto de desenvolvimento de Software, cujo foco do programa é o Modelo de Referência MPS para Software (MR-MPS-SW), desenvolvido pela Softex (2016). Como pode-

se observar, este Modelo é apresentado no Guia Geral MPS de Software, e nele, é descrito o Modelo de Referência MPS-SW, definindo os níveis de maturidade MPS, os processos relacionados a Software, com seus propósito e resultados esperados e os atributos de processo, que definem o nível de capacidade dos processos esperada em cada nível de maturidade. (SOFTEX, 2016).

Este Guia é complementado pelos Guias de Implementação do MR-MPS-SW, que fornecem uma fundamentação teórica e informações complementares para apoio à implementação dos níveis de maturidade.

Além disso, as avaliações do MR-MPS-SW devem ser realizadas de acordo com o método de avaliação MPS (MA-MPS) descrito no Guia de Avaliação do MPS. (SOFTEX, 2017).

2.4 O MODELO DE REFERÊNCIA MPS PARA SOFTWARE (MR-MPS-SW)

Esse modelo tem como base técnica a ISO/IEC 12207 e o CMMI-DEV. Dessa forma, os níveis de maturidade são uma combinação entre processos e sua capacidade. (SOFTEX, 2016). Ademais, segue os requisitos para um modelo de referência de processo apresentados na ISO/IEC 33020, declarando o propósito e os resultados esperados de sua execução. (SOFTEX, 2016)

Segundo a ISO/IEC 33020 (2008), avaliar o nível de maturidade através dos resultados esperados permite atribuir graus de efetividade na execução dos processos. Assim as atividades e tarefas necessárias para atender esses resultados esperados não são definidas no modelo, devendo ficar a cargo dos usuários do MR-MPS-SW. (SOFTEX, 2016)

Entendidos os resultados esperados de processo, o Guia Geral MPS de Software da Softex (2016, pag. 17) define:

“A capacidade do processo é a caracterização da habilidade do processo para alcançar os objetivos de negócio, atuais e futuros, estando relacionada com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade. Essa capacidade é representada por um conjunto de atributos de processo, o qual, expressa o grau de refinamento e institucionalização com que o processo é executado na organização/unidade organizacional.”

Tabela 1 - Níveis de maturidade do MR-MPS-SW.

Nível	Processos	Atributos de Processo
A	-	AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Riscos - GRI Desenvolvimento para Reutilização – DRU Gerência de Decisões – GDE	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
D	Verificação – VER Validação – VAL Projeto e Construção do Produto – PCP Integração do Produto – ITP Desenvolvimento de Requisitos – DRE	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
E	Gerência de Projetos – GPR (evolução) Gerência de Reutilização – GRU Gerência de Recursos Humanos – GRH Definição do Processo Organizacional – DFP Avaliação e Melhoria do Processo Organizacional – AMP	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
F	Medição – MED Garantia da Qualidade – GQA Gerência de Portfólio de Projetos – GPP Gerência de Configuração – GCO Aquisição – AQU	AP 1.1, AP 2.1 e AP 2.2
G	Gerência de Requisitos – GRE Gerência de Projetos – GPR	AP 1.1 e AP 2.1

Fonte: Softex (2016).

Logo, deve-se avaliar a capacidade de processo a partir dos Atributos de Processo (AP). Também, é destacado no próprio Guia Geral MPS de Software (2016), que o atendimento de cada atributos do processo é descrito como requerido para cada um dos processos no nível correspondente, embora eles não sejam detalhados dentro de cada processo.

Além disso, o nível de capacidade deve progredir junto ao nível de maturidade. Logo, organizações com um maior nível de maturidade deve atingir um maior nível de capacidade para desempenhar seu processo. (SOFTEX, 2016).

Os distintos níveis de capacidade dos processos são delineados por APs, sendo que a conquista de cada AP é avaliada utilizando resultados que indicam a implementação completa do atributo, chamados Resultados de Atributos de Processo (RAP). (SOFTEX, 2016).

Assim, a Tabela 1 colhida do Guia Geral MPS de Software (2016) apresenta os níveis de maturidade do MR-MPS-SW, os processos e os atributos de processo correspondentes a cada nível.

Conclui-se, portanto, que o modelo fornece uma lista de processos e seus resultados esperados, os Atributos de Processos e seus Resultados de Atributos de Processo esperados, como referência para implantação nas organizações de desenvolvimento de software.

2.4.1 O nível G do MR-MPS-SW

O nível G é o primeiro nível de maturidade do MR-MPS-SW e é por ele que se deve iniciar uma implantação do modelo. Para atingi-lo, deve se atender aos resultados esperados de Gerência de Requisitos (GRE), aos resultados esperados de Gerência de Projetos (GPR) e aos Atributos de Processo AP 1.1 e AP 2.1. Ao final da implantação deste nível a organização deve ser capaz de gerenciar parcialmente seus projetos de desenvolvimento de software. (SOFTEX, 2016).

De acordo com o Guia Geral do MR-MPS-SW, segue a definição desses dois APs e seus respectivos RAPs (SOFTEX, 2016):

- AP 1.1 O processo é executado: é a medida do quanto o propósito do processo é alcançado pela sua execução. Como RAP:
 - (i) O processo produz os resultados definidos.
- AP 2.1 A execução do processo é gerenciada: é a medida do quanto a execução do processo é gerenciada. Como RAP:
 - (i) existe uma política organizacional estabelecida e mantida para o processo;

- (ii) a execução do processo é planejada (O planejamento deve incluir identificação e disponibilização dos recursos e informações necessárias para a execução do processo, definição, atribuição e comunicação das responsabilidades pela execução do processo e planejamento da comunicação entre as partes interessadas);
- (iii) a execução do processo é monitorada em relação ao planejado e, quando necessário, ajustes são realizados;
- (iv) as pessoas que executam o processo estão preparadas para executar suas responsabilidades;
- (v) as atividades, o status e os resultados do processo são revistos com a gerência de nível superior e são tratadas questões críticas. (SOFTEX, 2016).

Entende a Softex (2016), portanto, que para o nível G, cada processo de GPR e de GRE do caso de estudo deve ter todos seis os citados RAP atendidos para ser considerados executados e gerenciados. As atividades e tarefas que alcançam esses resultados ficam a cargo da organização, desde que cumpram os RAPs. Cada GPR e GRE será apresentada a seguir.

2.4.1.1 Gerência de Projetos (GPR)

A Softex (2016, pag. 24) define que:

“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 *Project Management Institute* define em seu site (PMI, 2019), gerenciamento de projeto como:

“[...] a aplicação de conhecimentos, habilidades e técnicas para a execução de projetos de forma efetiva e eficaz. Trata-se de uma competência estratégica para organizações, permitindo com que elas unam os resultados dos projetos com os objetivos do negócio – e, assim, melhor competir em seus mercados.”

Dentre as atividades de GPR, a organização deve desenvolver um plano geral de projeto, contendo: estimativa de escopo, produtos de trabalho (entregáveis) e tarefas do projeto com respectivo cronograma, seus recursos e dos riscos inerentes. É também importante para o GPR acompanhar o avanço do projeto, comparando os itens do plano geral com o que foi efetivamente executado. (SOFTEX, 2016)

Acerca disso, são apresentados os resultados esperados para o processo Gerência de Projetos conforme o nível G da Softex (2016, pag. 24 a 26):

- “GPR 1. O escopo do trabalho para o projeto é definido;
- GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;
- GPR 3. O modelo e as fases do ciclo de vida do projeto são definidos;
- GPR 4. (Até 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;
- GPR 5. O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos;
- GPR 6. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;
- GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;
- GPR 8. (Até o nível F) Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;
- GPR 9. Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;
- GPR 10. Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos;
- GPR 11. A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e

recursos disponíveis. Se necessário, ajustes são realizados;

- GPR 12. O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido;
- GPR 13. O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado;
- GPR 14. Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado;
- GPR 15. Os riscos são monitorados em relação ao planejado;
- GPR 16. O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido;
- GPR 17. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;
- GPR 18. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;
- GPR 19. Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão;”

No Guia de Implantação, estão destacadas observações que procuram explicitar elementos importantes na interpretação desses GPRs. Essas observações foram resumidas e inseridas no Anexo B. Vale ressaltar que o atendimento dessas observações e exemplos oferecidos não é obrigatório.

No Guia de Avaliação (SOFTEX, 2017), é explicado que, durante uma avaliação MPS.BR, é requerido somente o atendimento aos GPRs definidos acima, extraídos do Guia Geral. (SOFTEX, 2016). Cabe, portanto, ao avaliador analisar se a implementação dos processos na organização atende a cada resultado, com abertura a múltiplas formas válidas de implementação. (SOFTEX, 2016).

2.4.1.2 Gerência de Requisitos (GRE)

Quanto ao GRE, a Softex (2016, pag. 27) define:

“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.”

Segundo o modelo MR-MPS-SW (SOFTEX, 2016), outras obrigações do GRE são: documentar as mudanças nos requisitos e suas justificativas; manter a rastreabilidade bidirecional entre os requisitos e produtos de trabalho em geral; e, identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto.

Segundo SEI (2010), requisitos bem gerenciados permitem o estabelecimento de rastreabilidade, desde a solicitação do cliente, passando pela decomposição do produto, até seus requisitos de mais baixo nível. Tal rastreabilidade, chamada de bidirecional, apoia na visualização de se todos os requisitos fonte foram atendidos.

Para tanto, a Softex (2016, pag. 27) traz os principais resultados esperados para o processo Gerência de Requisitos conforme o nível G:

- “GRE 1. O entendimento dos requisitos é obtido junto aos fornecedores de requisitos;
- GRE 2. Os requisitos são avaliados com base em critérios objetivos e um comprometimento da equipe técnica com estes requisitos é obtido;
- GRE 3. A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
- GRE 4. Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
- GRE 5. Mudanças nos requisitos são gerenciadas ao longo do projeto.”

No Guia de Implantação, também estão destacadas observações que procuram explicitar elementos importantes na interpretação desses GREs. Essas observações foram resumidas e inseridas no Anexo B. Vale ressaltar que o atendimento dessas observações e exemplos oferecidos não é obrigatório.

No Guia de Avaliação (SOFTEX, 2017), é explicado que, durante uma avaliação MPS.BR, é requerido somente o atendimento aos GREs definidos acima, extraídos do Guia Geral. (SOFTEX, 2016). Cabe,

portanto, ao avaliador analisar se a implementação dos processos na organização atende a cada resultado, com abertura a múltiplas formas válidas de implementação. (SOFTEX, 2016).

2.4.2 Práticas e ferramentas utilizadas

Como comentado ao apresentar o MR-MPS-SW, cabe à empresa determinar as atividades e tarefas que alcançam esses resultados de GPR, GRE e RAP. A empresa do estudo de caso estudado declarou que seus processos utilizam primordialmente o método Scrum como base para delimitar seus processos, com apoio de conceitos do PMBOK (*Project Management Body of Knowledge*) e RUP (*Rational Unified Process*). Dessa forma, esses métodos serão apresentados a seguir.

Além disso, para a análise das propostas de melhorias oferecidas, utilizou-se da ferramenta 5W2H, que fornece uma visão clara das alterações propostas.

2.4.2.1 Scrum

O Scrum destaca-se como uma das formas válidas de implementação. Segundo o Guia Scrum (SCHWABER e SUTHERLAND, 2017), podemos definir o Scrum como:

Uma estrutura na qual as pessoas podem lidar com problemas complexos de adaptação, ao mesmo tempo em que produzem de maneira produtiva e criativa produtos com o maior valor possível.

O Scrum pode ser classificado como uma prática ágil, leve simples de entender, porém difícil de dominar. (SCHWABER e SUTHERLAND, 2017). Segundo Lima e Vendramel (2011), é adaptativo e empírico, ou seja, possui maior flexibilidade no desenvolvimento de software, ao contrário dos métodos tradicionais.

Scrum não é um processo, técnica ou método definitivo, mas é uma estrutura na qual você pode empregar vários processos e técnicas. O Scrum deixa clara a eficácia relativa do gerenciamento de produtos e das técnicas de trabalho para que você possa melhorar continuamente o produto, a equipe e o ambiente de trabalho. (SCHWABER e SUTHERLAND, 2017).

Schwaber e Sutherland (2017) explicam que a estrutura do Scrum consiste em equipes do Scrum e suas funções, eventos, artefatos e regras

associados. Cada componente da estrutura serve a um propósito específico e é essencial para o sucesso e uso do Scrum. Três pilares sustentam todas as implementações do controle empírico de processos: transparência, inspeção e adaptação.

A seguir, são listadas as funções no Scrum (SCHWABER e SUTHERLAND, 2017):

1. *Scrum Team*: é a Equipe Scrum, consiste em 3 funções, um *Product Owner*, um *Scrum Master* e um *Development Team*. As equipes Scrum são auto organizadas e multifuncionais.
 - a. *Product Owner*: é o dono do Produto, responsável por maximizar o valor do produto resultante do trabalho da Equipe de Desenvolvimento.
 - b. *Scrum Master*: é um líder-servidor da equipe Scrum. é responsável por promover e apoiar o Scrum.
 - c. *Development Team*: a equipe de desenvolvimento é composta por profissionais que realizam o trabalho de fornecer um incremento potencialmente liberável do produto "Concluído" no final de cada *Sprint*, é recomendado ser equipes de 3 a 9 membros.

Schwaber e Sutherland (2017) definem os eventos Scrum:

2. *Sprint*: é o tempo de ciclo do desenvolvimento de um incremento. Os *sprints* têm durações variáveis, sendo um intervalo de um mês ou menos. A duração mais utilizada é de 2 semanas.
 - a. *Sprint Planning*: O trabalho a ser executado no *Sprint* é planejado no Planejamento do *Sprint*. Esse plano é criado pelo trabalho colaborativo de toda a equipe Scrum.
 - b. *Daily Scrum*: também conhecido como *Daily Meeting*, é um evento de 15 minutos realizado todos os dias do *Sprint*. Nele, a equipe de desenvolvimento planeja o que será trabalhado nas próximas 24 horas.
 - c. *Sprint Review*: uma Revisão da *Sprint* é realizada no final da *Sprint* para inspecionar o Incremento e adaptar o Backlog do Produto.
 - d. *Sprint Retrospective*: A Retrospectiva da *Sprint* é uma oportunidade para a Equipe Scrum se inspecionar e criar um plano para melhorias a serem implementadas durante a próxima *Sprint*.

A Figura 7 ilustra como é o processo iterativo desse modelo de ciclo de vida.

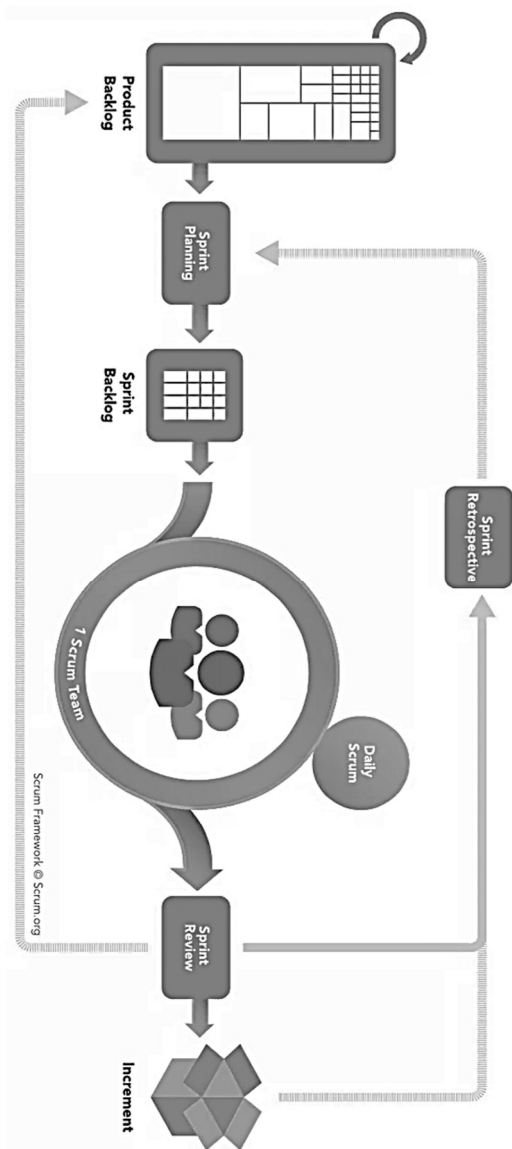


Figura 7 - Fluxo de desenvolvimento de um incremento.
Fonte: Scrum Portugal (2017).

Como pode-se observar, alguns artefatos são inerentes desse modelo. Schwaber e Sutherland (2017) definem os artefatos específicos do Scrum:

3. *Product Backlog*: o *Backlog* do produto é uma lista ordenada de tudo o que se sabe ser necessário no produto. É a única fonte de requisitos para quaisquer alterações a serem feitas no produto. O dono do Produto é responsável pelo *Backlog* do Produto, incluindo seu conteúdo, disponibilidade e pedido.
4. *Sprint Backlog*: o *Backlog* da *Sprint* é o conjunto de itens do *Backlog* do produto selecionados para o *Sprint*, além de um plano para fornecer o incremento do produto e atingir a meta do *Sprint*.
5. Incremento: é o conjunto dos itens do *Backlog* do produto concluídos durante uma *Sprint*. É um passo em direção a uma visão ou objetivo.

As regras do Scrum unem os papéis, eventos e artefatos, governando os relacionamentos e a interação entre eles.

2.4.2.2 Guia PMBOK

O *Project Management Institute* (PMI, 2017) define o Conhecimento em Gerenciamento de Projetos (PMBOK – *Project Management Body of Knowledge*) como um termo que remete a todo o conhecimento relacionado gerenciamento de projetos existente. Isso inclui práticas tradicionais comprovadas e amplamente aplicadas e práticas novas surgindo na profissão.

O Guia PMBOK®, por sua vez, é um subconjunto de todo esse conhecimento. O guia busca apresentar um compilado de conhecimentos em gerenciamento de projetos geralmente reconhecidos como boas práticas. Ele é mantido e publicado pelo PMI (*Project Management Institute*), um dos institutos internacionais mais conceituados e reconhecidos na área de gerenciamento de projetos. (PMI, 2017).

Logo, O Guia PMBOK fornece boas práticas e diretrizes para apoiar o gerenciamento de projetos individuais, não tratando de gerenciamento de portfólios, e apresenta os conceitos relacionados com o gerenciamento de projetos. (PMI, 2017).

2.4.2.3 RUP

O *Rational Unified Process* (RUP) é um processo de engenharia de software criado pela *Rational Software Corporation* e oferecido atualmente pela IBM (*International Business Machines Corporation*), uma das maiores empresas dos Estados Unidos voltada para a área de informática. (RATIONAL, 2001).

O RUP é um compilado de melhores práticas para equipes de desenvolvimento de software, buscando refletir experiências e melhores práticas em evolução e comprovadas. É adequado para uma ampla gama de projetos e organizações. Implementando essas práticas recomendadas usando o RUP como seu guia oferece às equipes de desenvolvimento várias vantagens importantes. (RATIONAL, 2001)

O RUP fornece uma abordagem disciplinada para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. Seu objetivo é garantir a produção de alta qualidade software que atenda às necessidades de seus usuários finais, dentro de um cronograma e orçamento previsíveis. (RATIONAL, 2001).

No RUP também foi introduzida a Linguagem de Modelagem Unificada (UML – *Unified Modeling Language*). A UML é uma linguagem padrão do setor de desenvolvimento de software que nos permite comunicar claramente os requisitos, arquiteturas e projetos. (RATIONAL, 2001).

2.4.2.4 Comparação entre o Guia PMBOK, RUP e Scrum

O Guia PMBOK trata de gerenciamento de projetos, porém trata de características específicas do desenvolvimento de software. Já o RUP é um ótimo modelo de processo que supre as necessidades para desenvolver softwares, porém não aborda custos ou recursos humanos. (MATSUSHITA, 2010).

O Scrum, por sua vez, é adaptativo e empírico, mas não atende todas as necessidades de organizações de software, como rastreabilidade bidirecional de requisitos e registro e acompanhamento de dados históricos. (LIMA, 2011).

Estas constatações indicam que eles são complementares. Deve-se avaliar os casos específicos para a definição de processo das organizações, utilizando um desses métodos ou uma solução híbrida. (LIMA e SOUSA, 2018).

2.4.3 Ferramenta 5W2H

SEBRAE (2019), em seu Plano de Ação para Empreendedores, define o 5W2H como uma ferramenta indicada a tomada de decisão sobre os principais elementos que orientarão a implementação do plano.

Quando algo deve ser implantado, deve-se responder as sete perguntas da ferramenta, apresentadas na Tabela 2. (SEBRAE, 2019).

Tabela 2 – Ferramenta 5W2H.

Perguntas da ferramenta		Tradução das perguntas	Respostas
5W	<i>What?</i>	O que? O que deve ser feito?	Ação, problema, desafio
	<i>Why?</i>	Por que? Porque deve ser implementado?	Justificativa, explicação, motivo
	<i>Who?</i>	Quem? Quem é o responsável pela ação?	Responsável
	<i>Where?</i>	Onde? Onde deve ser executado?	Local
	<i>When?</i>	Quando? Quando deve ser implementado?	Prazo, cronograma
2H	<i>How?</i>	Como? Como deve ser conduzido?	Procedimentos, etapas
	<i>How much?</i>	Quanto? Quanto vai custar a implementação?	Custo, desembolsos

Fonte: Softex (2016).

O 5W2H, também conhecida como plano de ação, é uma ferramenta simples e ganhou mais popularidade com a disseminação das

técnicas de gestão da qualidade e, posteriormente, com as de gestão de projetos. (SEBRAE, 2019).

2.5 MODELO DE AVALIAÇÃO MPS (MA-MPS)

O Guia de Avaliação descreve o Processo e Método de Avaliação MA-MPS, os requisitos para avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IA). (SOFTEX, 2017).

O MA-MPS é separado em duas partes (SOFTEX, 2017):

- Parte 1: descreve o Processo e o Método de Avaliação MA-MPS. É baseado na série de Normas Internacionais 33000, principalmente a ISO/IEC 33020: 2015;
- Parte 2: descreve os requisitos para a constituição de Instituições Avaliadoras e para o credenciamento de avaliadores MA-MPS.

Como o escopo do trabalho não inclui certificação ou credenciamento de avaliadores ou instituições avaliadoras, voltou-se o estudo para a Parte 1 do guia (SOFTEX, 2017). Nela, são descritas as etapas de avaliação e considerações sobre essas etapas. Entre seus objetivos, destacam-se:

- “Permitir a avaliação objetiva dos processos de software, de serviços ou de gestão de pessoas de uma organização/unidade organizacional” (SOFTEX, 2017);
- “Permitir a atribuição de um nível de maturidade do MR-MPS-SW, MR-MPS-SV ou MR-MPS-RH com base no resultado da avaliação” (SOFTEX, 2017).

Softex (2017) destaca que, para que uma avaliação seja conduzida com sucesso, é necessário comprometimento, motivação, fornecimento de feedback e percepção dos benefícios por parte da organização avaliada. Além disso, deve-se ter um comprometimento do avaliador com a credibilidade da avaliação e confidencialidade das informações obtidas.

O processo de avaliação é dividido em 4 subprocessos: preparar a realização da avaliação; realizar a avaliação inicial; realizar a avaliação final e documentar os resultados da avaliação. Cada subprocesso do processo de avaliação é composto de atividades. Cada atividade é descrita por meio das tarefas que a compõem. (SOFTEX, 2017).

Vale destacar que, muitas dessas atividades e tarefas estão relacionadas diretamente com a certificação. Logo, filtrou-se as etapas

relevantes para o estudo de caso, que foca apenas em melhoria interna do projeto. Além disso, considerações importantes sobre algumas dessas atividades e tarefas, como seleção das pessoas a serem entrevistadas, a coleta de indicadores, realização das entrevistas, foram compiladas no Anexo A.

O Guia de Avaliação – Parte 1 (SOFTEX, 2017) fornece também as escalas para caracterização do grau de implementação de cada resultado esperado de processo e dos RAPs: Totalmente implementado (T), Largamente implementado (L), Parcialmente implementado (P) e Não implementado (N). A Tabela 3 apresenta a caracterização do grau de implementação de cada GPRs e GREs.

Tabela 3 – Escala para caracterização do grau de implementação de um resultado esperado do processo nos projetos/serviços/área

Grau de implementação	Caracterização
Totalmente implementado (T)	<ul style="list-style-type: none"> - O indicador direto está presente e é julgado adequado - Existe pelo menos uma afirmação confirmando a implementação - Não foi notado nenhum ponto fraco substancial na avaliação inicial ou na avaliação final.
Largamente implementado (L)	<ul style="list-style-type: none"> - O indicador direto está presente e é julgado adequado - Existe pelo menos uma afirmação confirmando a implementação - Foi notado um ou mais pontos fracos substanciais na avaliação inicial ou na avaliação final.
Parcialmente implementado (P)	<ul style="list-style-type: none"> - O indicador direto não está presente ou é julgado inadequado - Artefatos/afirmações sugerem que alguns aspectos do resultado esperado estão implementados - Foi notado um ou mais pontos fracos substanciais.
Não implementado (N)	<ul style="list-style-type: none"> - Qualquer situação diferente das acima
Não avaliado (NA)	<ul style="list-style-type: none"> - O projeto/serviço/área não está na fase de desenvolvimento que permite atender ao resultado ou não faz parte do escopo do projeto atender ao resultado.
Fora do escopo (F)	<ul style="list-style-type: none"> - O resultado esperado está fora do escopo da avaliação, conforme documentado no plano da avaliação.

Fonte: Softex (2017).

Quantos aos atributos de processo e seus RAPs, a caracterização do grau de implementação de cada resultado esperado segue os parâmetros estabelecidos na Tabela 4.

Tabela 4 – Regras para caracterizar o grau de implantação dos atributos do processo na unidade organizacional

Grau de implementação	Caracterização	Porcentagem de implementação dos resultados relacionados
Totalmente implementado (T)	Existe evidência de um enfoque completo e sistemático para o atributo no processo avaliado e de sua plena implementação. Não existem pontos fracos relevantes para este atributo no processo avaliado.	>85% a 100%
Largamente implementado (L)	Existe evidência de um enfoque sistemático e de um grau significativo de implementação do atributo no processo avaliado. Existem pontos fracos para este atributo no processo avaliado.	>50% a 85%
Parcialmente implementado (P)	Existe alguma evidência de um enfoque para o atributo e de alguma implementação do atributo no processo avaliado. Alguns aspectos de implementação não são possíveis de prever.	>15% a 50%
Não implementado (N)	Existe pouca ou nenhuma evidência de implementação do atributo no processo avaliado	0 a 15%

Fonte: Softex (2017).

Após caracterizar o grau de implementação de cada resultado de processo e de cada atributo de processo na unidade organizacional, é caracterizado o grau de implementação de cada processo como SATISFEITO ou NÃO SATISFEITO. Logo, os considerados NÃO SATISFEITO, serão o foco na etapa de proposição de melhoria.

Vale ressaltar que um processo está SATISFEITO quando:

- Todos os resultados esperados para o processo foram caracterizados como T (Totalmente implementado) ou L (Largamente implementado).
- A caracterização dos atributos de processo satisfaz às exigências da Tabela 5. Em qualquer outra situação o processo é caracterizado como NÃO SATISFEITO.

Tabela 5 – Caracterização de atributos do processo para satisfazer aos níveis MR-MPS

Nível MR-MPS	Atributos do processo	Caracterização
G	AP 1.1	T
	AP 2.1	L ou T

Fonte: Softex (2017).

Por fim, há uma restrição específica para o AP 1.1: para que esse atributo seja considerado Totalmente implementado (T), nenhum dos processos pode estar classificado nesse atributo como “Não implementado”. Ou seja, todos os processos listados para o Nível G devem ser executados.

2.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foi realizada uma revisão da literatura no tocante aos conceitos relacionados aos processos produtivos de desenvolvimento de software, melhoramento de processos de desenvolvimento de software e avaliação desse processo.

Também, foi apresentado o programa MPS.BR e seu modelo de referência para desenvolvimento de softwares, o MR-MPS-SW. O nível G desse modelo será utilizado como base para elaboração da metodologia de avaliação e caracterização do processo do caso estudado.

Ademais, foram apresentados o Guia PMBOK, RUP e SCRUM, que serão consultados posteriormente como fonte para a proposição de formas de alcançar os resultados esperados para o nível G. Também, apresentou-se o 5W2H, técnica utilizada para analisar as propostas de melhorias realizadas.

Por fim, foi apresentado o MA-MPS, base para o plano de avaliação realizado.

3 METODOLOGIA

Neste capítulo foi apresentada a caracterização científica desse trabalho, suas etapas metodológicas e delimitações. Além disso, será detalhado o método de avaliação e proposição de melhorias utilizados.

3.1 CARACTERIZAÇÃO CIENTÍFICA DO TRABALHO

A natureza dessa pesquisa é aplicada, já que se trata de uma utilização prática de metodologia de avaliação de processos de desenvolvimento de software para identificar os problemas de um projeto.

A forma de abordar o problema é predominantemente qualitativa, apesar da utilização de escalas de caracterização, já que se trata de uma análise apresentando informações levantadas em entrevista e análise documental.

Além disso, a abordagem também apresenta características exploratórias ao realizar o levantamento bibliográfico relacionado ao tema. Por fim, a abordagem também apresenta características descritivas, já que será descrito o processo de desenvolvimento em vigor no projeto estudado.

Do ponto de vista dos procedimentos técnicos, este estudo se caracteriza como um estudo de caso, diagnosticando a fundo o processo de desenvolvimento de software de um projeto específico.

3.2 ETAPAS METODOLÓGICAS DO TRABALHO

Nesta subseção, explica-se como foi realizado o estudo. A Figura 8, ilustra essas etapas.

Primeiramente, define-se o tema, a questão da pesquisa e os objetivos, apresentados na introdução. Em seguida, realiza-se uma pesquisa bibliográfica relacionada ao melhoramento de processos de desenvolvimento de software. Com essa pesquisa, pode-se conhecer as principais técnicas utilizadas atualmente pelas empresas do país.

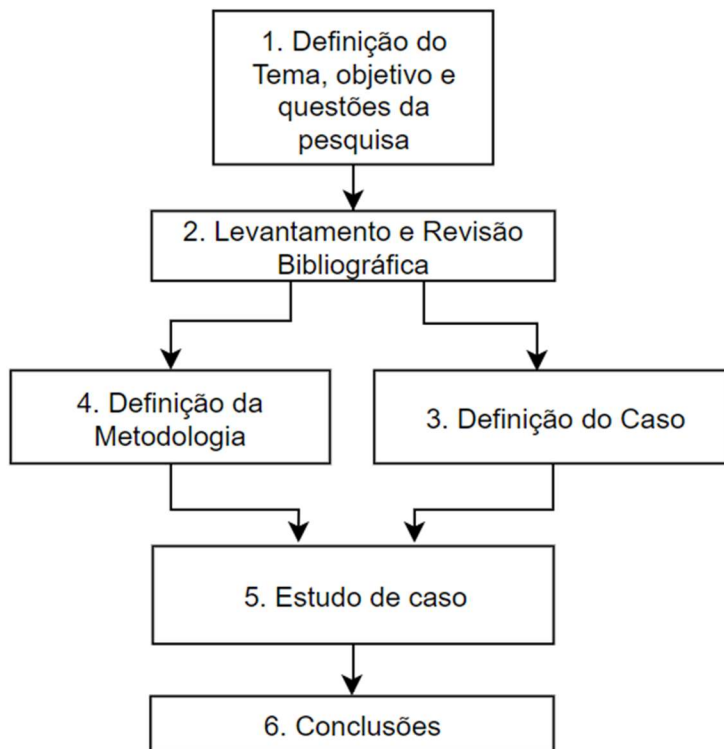


Figura 8 – Etapas metodológicas do trabalho
Fonte: elaborado pelo autor.

Em seguida, definiu-se o caso para o estudo, sendo disponibilizado um projeto de desenvolvimento de um repositório para educação de trânsito. A organização em que foi realizado o estudo é especializada na criação de sistemas customizados, projetados conforme o modelo de negócio do cliente, relacionados ao setor transporte e logística.

Com a metodologia e o caso do trabalho definidos, foi possível realizar a avaliação. A forma utilizada na avaliação foi detalhada no item 3.3 desse trabalho. Após a avaliação, os resultados foram analisados e melhorias propostas.

Por fim, tirou-se conclusões sobre o estudo de caso. O comprometimento com a execução futura das melhorias ficou a cargo da equipe do projeto.

3.3 MÉTODO DE AVALIAÇÃO UTILIZADO NO ESTUDO DE CASO

O método utilizado na avaliação e proposta de melhoria baseou-se no “Guia de Avaliação - Parte I – Processo e Método de Avaliação MA-MPS” e no “Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS-SW:2016”. As etapas de coleta de dados, análise do material e relatório, realizou-se de acordo com o plano de avaliação. A sequência de tarefas planejada para a avaliação foi adaptada para esse trabalho do MA-MPS, visualiza-se na Figura 9.

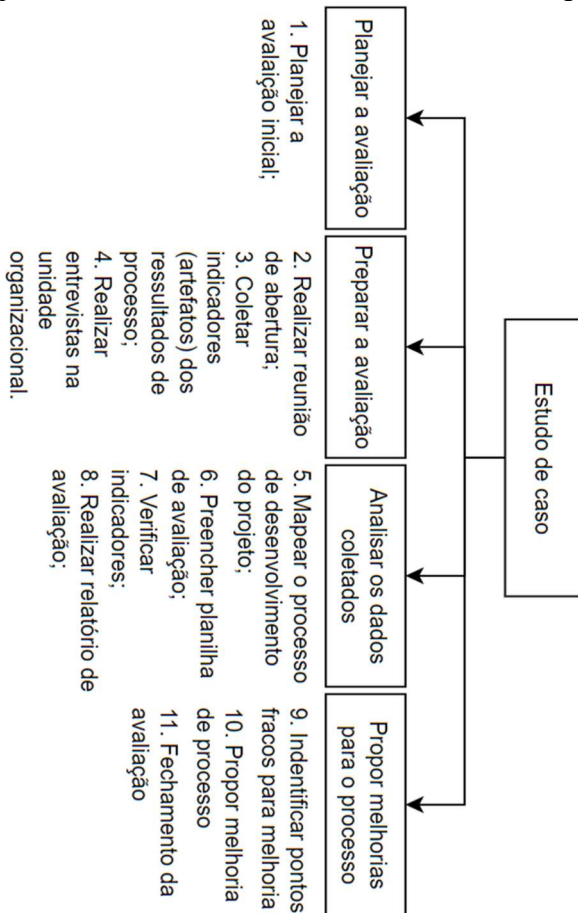


Figura 9 - Planejamento da avaliação
Fonte: elaborado pelo autor.

Nele, como se pode observar, o plano de avaliação anterior é composto por 4 (quatro) processos, quais sejam: planejar a avaliação, preparar a avaliação, analisar os dados coletados e propor melhorias para o processo, cada qual com sua especificidade de atuação, devidamente adaptado ao estudo corrente.

A fim de obter-se um panorama inicial da situação em que se encontram os processos de Gerenciamento de Projetos e Requisitos da instituição, executou-se uma avaliação da forma de geração e documentação dos artefatos produzidos durante o desenvolvimento de softwares.

Para tal, coletou-se uma série de documentos gerados ao longo do processo para uma análise documental e como evidências da execução do processo, chamados pelo Guia de Avaliação de indicadores diretos do processo (SOFTEX, 2017). Forneceu-se, pela empresa, acesso a ferramenta utilizada para armazenamento e compartilhamento no projeto de documentos, como o Plano de trabalho e Análise Preliminar de Demanda. Forneceu-se, também, acesso a base de dados de documentação de análise de sistemas, acessada através um software específico para gestão de produto, o *Enterprise Architect*.

Quanto a ferramenta de gestão de atividades de desenvolvimento, chamada de Gitlab, o acesso também foi fornecido, permitindo visualização das *User Storys*, suas atividades e a metodologia de trabalho. Ainda, o pesquisador participou de diversos ciclos de *Sprints*, e, de todos os eventos *Scrum*, desde *Daily Meetings* até *Sprints Planing, Review e Retrospective*.

Já as entrevistas foram realizadas com 5 dos 8 integrantes da equipe: 1 analista de sistemas, 2 desenvolvedores, 1 design e 1 tester. Buscou-se realizar as entrevistas com o máximo de pessoas possível e utilizando as regras expostas no Anexo A, extraídas do Guia de Avaliação do MPS.BR. (SOFTEX, 2017). Notas foram tomadas para posterior consulta e para mapeamento do processo.

Além disso, para orientar as entrevistas, um questionário foi elaborado, com perguntas criadas utilizando como referência o Apêndice A, baseado nos resultados esperados expostos no Guia de Implementação. (SOFTEX, 2016).

Essas experiências, informações e documentos foram insumos para preenchimento de uma planilha para tratamento dos dados e avaliação, previamente preparada e baseada nos parâmetros para alcance do nível G no MR-MPS-SW. O nível G no MR-MPS-SW introduz os resultados esperados do processo para GPR e GRE, já apresentados na

fundamentação teórica. Os detalhamentos quanto a esses resultados podem ser vistos no Anexo B.

As escalas de caracterização utilizadas foram as mesmas do MA-MPS, já apresentadas na fundamentação teórica: Totalmente implementado (T), Largamente implementado (L), Parcialmente implementado (P) e Não implementado (N). Também, os critérios de satisfação do nível G foram os mesmos do MA-MPS.

Cada GPR e GRE foi detalhado em relação aos seus resultados esperados e, para cada resultado esperado, foram apresentadas formas de atendimento. Para encontrar essas formas de atendimento, restringiu-se às práticas do Scrum, Guia PMBOK e RUP.

3.4 DELIMITAÇÕES DO TRABALHO

Este trabalho se limita a apontar as oportunidades de melhoria dos processos já existentes com base no nível G de maturidade do modelo MPS.BR em um projeto piloto. Caso a empresa considere realizar as alterações e que as alterações testadas nesse projeto sejam vantajosas, essas práticas poderão ser gradativamente implantadas nos outros projetos. No entanto, não há o compromisso da empresa de cumprir todas as práticas, nem obter uma certificação no nível G do MPS.BR.

Foi escolhido o nível G do MPS.BR por ser o primeiro nível. Os níveis são acumulativos, tendo sentido escolher realizar a primeira avaliação do projeto a partir do primeiro degrau do modelo. Ademais, o Guia de Avaliação salienta que só é recomendado avançar aos níveis superiores com os níveis inferiores atendidos.

As melhorias foram propostas para o projeto piloto como experimentação, com o apoio da equipe e supervisão da liderança do projeto e liderança técnica da equipe de desenvolvimento. Espera-se observar ao final desse trabalho a realização da implantação dessas propostas. Buscou-se conscientizá-los, da melhor forma possível, da importância das melhorias. Porém, não será responsabilidade do pesquisador a executar as tarefas de melhoria previstas no modelo.

A organização do estudo de caso determinou que baseia sua metodologia de trabalho no Scrum, com algumas práticas do Guia PMBOK e RUP. Dessa forma, considerando essa declaração e considerando a melhor chance de aceitação das propostas, as soluções apresentadas neste trabalho foram buscadas nesses conjuntos de práticas.

4 O ESTUDO DE CASO

Neste capítulo foi primeiramente apresentados o ambiente do estudo de caso e o processo mapeado do projeto. Em seguida, são feitas considerações quanto aos GPR e GRE desse processo, destacando os pontos fortes e pontos fracos do processo. Enfim, foi apresentado o resultado da avaliação, de acordo com o nível G do MR-MPS-SW.

Em seguida, são realizadas as propostas de melhorias. Essas propostas são detalhadas e, por fim, são analisadas utilizando a técnica 5W2H.

4.1 O AMBIENTE DO ESTUDO DE CASO

Atualmente, a empresa tem um total de, aproximadamente, 80 colaboradores em seu quadro de funcionários. Por motivos internos, a empresa solicitou anonimato nesse trabalho, que será respeitado. Apenas informações necessárias do processo foram permitidas utilização, porém, detalhamentos dos produtos serão mantidos confidenciais.

O estudo de caso foi realizado em um projeto em andamento, determinado na empresa como um projeto piloto. A empresa considera um projeto piloto aquele em que são experimentadas novas ideias. No contexto desse trabalho, foi autorizado experimentar novos processos e novas ferramentas neste projeto, sob supervisão da liderança do projeto.

No projeto participam 8 colaboradores:

- 1 analista de negócio;
- 1 analista de sistemas;
- 2 desenvolvedores juniores;
- 1 desenvolvedor pleno;
- 1 desenvolvedor sênior;
- 1 designer;
- 1 testers;

O estudo de caso é no projeto de desenvolvimento de um repositório para conteúdo para educação de trânsito. Nesse repositório são inseridos projetos, cursos e atividades para serem utilizado em salas de aula.

Para compor esse repositório, são desenvolvidos 3 produtos: um site para acesso dos alunos e professores ao conteúdo do repositório; um aplicativo para dispositivos móveis como alternativas ao repositório para esses alunos e professores; e, um sistema gerenciador de conteúdo para o

cliente coordenar os projetos, cursos e atividades do site.

4.2 O PROCESSO DO PROJETO

A Figura 10 ilustra a etapa inicial do projeto, desde a definição do escopo até o fechamento do Plano de trabalho e contratação do serviço.

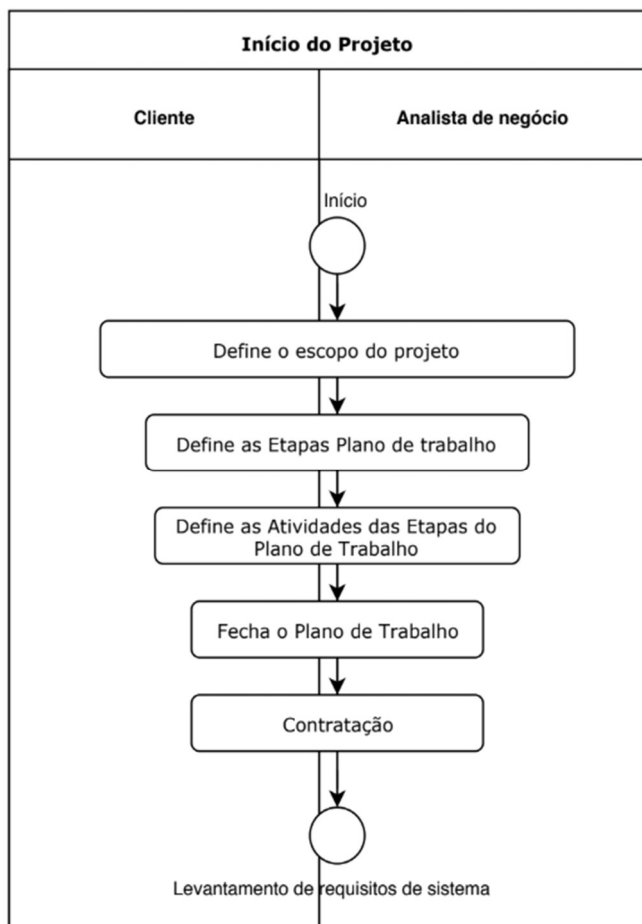


Figura 10 - Etapa de iniciação do projeto.
Fonte: elaborado pelo autor.

Inicialmente, antes do fechamento do contrato e estabelecimento do plano de trabalho, o analista de negócio realiza uma série de reuniões

e interações por e-mail, telefone ou videoconferência, onde é obtido o entendimento geral do negócio do cliente e das suas necessidades. A partir desse entendimento da necessidade da demanda do cliente, é formatada uma proposta comercial de trabalho com o escopo proposto.

Essa proposta é um documento formal que é enviado para o cliente, já com as configurações do Plano de Trabalho (Anexo D), para ser analisado pelo cliente.

Caso o cliente mostre interesse em seguir com as negociações, ir-se-á para a etapa de refinamento dessa proposta. As partes operacionais, táticas e estratégicas do cliente solicitam ajustes, adição e/ou remoção de alguns pontos desse documento. Com isso, são compreendidas as necessidades gerais. Nesse momento, portanto, é definido o escopo geral do projeto, os produtos, mas, até o momento, não há detalhamento das etapas.

Em seguida é acertado com o cliente como será executado o projeto, as interações entre as partes, o modelo de ciclo de vida estabelecido, como serão coletados dados relevantes para o projeto e outros aspectos.

Nesse momento, são estabelecidas, também, as chamadas Etapas de Trabalho inseridas no Plano de Trabalho, subdivididas em atividades. Essas atividades serão utilizadas para gerar um cronograma de atividades, orçamento, marcos e previsão de entregas. As estimativas do plano de trabalho serão utilizadas como base para o desenvolvimento. Porém, há uma possibilidade real de não serem mantidos esses cronogramas, já que o desenvolvimento de software customizado se mostra bastante imprevisível. De acordo com decisões estratégicas ou necessidades do projeto, esse cronograma é adaptável.

Tendo o Plano de trabalho do projeto consolidado, é fechada a contratação com o cliente e são iniciadas as atividades de desenvolvimento, começando pelo levantamento de requisitos do sistema (Figura 11).

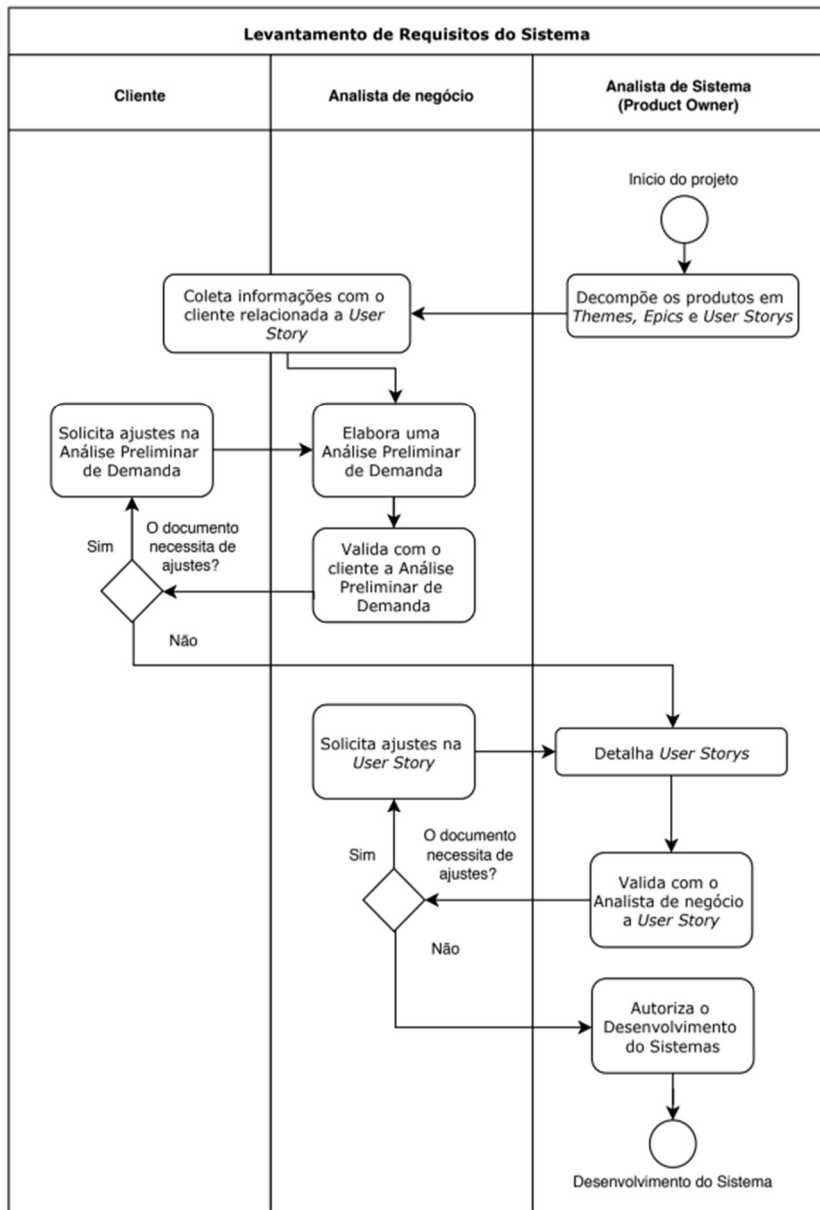


Figura 11 - Etapa de Levantamento de Requisitos do Sistema

Fonte: elaborado pelo autor.

Com o Plano de Trabalho, o analista de sistemas, decompõe os produtos em atividades com componentes cada vez menores, iniciando pelos *Themes*, que são quebrados em *Epics* e, por fim, *User Storys*.

Internamente, o desenvolvimento do projeto utiliza de um modelo de ciclo de vida próprio, baseado no *framework Scrum*, para desenvolver suas atividades.

O *Scrum Master* é o desenvolvedor que maior experiência, desenvolvedor sênior e o *Product Owner* é o analista de sistema. Os outros integrantes estão inseridos no papel da Equipe Dev. As *Sprints*, por sua vez, foram definidas em *time boxes* de 2 semanas.

As iterações ocorrem na esfera de especificação e execução do produto. Com a subdivisão em *User Storys* como ponto de partida, são iniciadas as atividades de detalhamento iterativo de funcionalidades do sistema.

O analista de negócio, é responsável por essa parte. Nela é realizado um estudo junto ao cliente e documentações disponíveis, a fim de identificar e entender melhor os objetivos de negócio do cliente de dos problemas a serem solucionados. Esses insumos são informações úteis para a concepção dos chamados de Requisitos de negócio.

No caso estudado, o analista de negócio e analista de sistemas têm acesso direto aos representantes do cliente, extraindo essas informações diariamente via telefone e e-mail. Além disso, uma vez por semana, nas quintas-feiras, são realizadas reuniões de duas horas para remoção de dúvidas em geral e alinhamento dessas informações. Esse formato de trabalho exige um cuidado extra com a consolidação dos insumos, caso contrário, pode resultar em informações incompletas ou distorcidas para as etapas posteriores do processo, comprometendo o desenvolvimento, que é voltado ao cliente.

Apesar da entrada ser variada, a saída desse processamento é bem estabelecida e consolidada em um documento padrão já existente: Análise Preliminar de Demanda (Anexo I), cujos principais pontos são os Requisitos do Negócio. Ajustar esse documento à especificidade do cliente é mostrar que há um comprometimento em também identificar as restrições, validações, condições e exceções do processo. Além disso, esse documento é validado com o cliente antes que siga no processo.

O analista de sistemas, com a Análise Preliminar de Demanda validada, passa a detalhar cada Requisito de negócio. Esse detalhamento não é mais com a visão do cliente e de negócio, mas com uma visão de solução atrelada a tecnologia que será desenvolvida, chamados de Requisitos de Sistema. Esse detalhamento é feito através da ferramenta *case* chamada *Enterprise Architect* e utilizando UML. Dentro da

ferramenta, definiu-se uma estrutura padrão de trabalho, apresentada no Anexo H.

O *Enterprise Architect* é um software que fornece uma série de funcionalidades de estruturação, versionamento e geração da documentação de análise de sistemas.

Vale ressaltar que esse detalhamento é feito aos poucos, de forma iterativas e incrementais ao longo do andamento do projeto, fornecendo gradativamente os insumos necessários para que a Equipe Dev possa planejar cada um de seus *Sprints*.

Esses requisitos de sistemas são atrelados às *User Storys* adequadas, para rastreamento. Para que siga no processo, essas *User Storys* e suas especificações são validadas pelo analista de negócio.

A próxima etapa, com os *User Storys* especificados e validados, a Equipe Dev pode iniciar a etapa de execução do desenvolvimento desse sistema (Figura 12).

Primeiramente, a Equipe Dev, em reunião de *Sprint Planning*, transforma essa demanda de *User Storys* em tarefas, as estimas e as registra em uma ferramenta de gerenciamento de atividades de desenvolvimento, chamada Gitlab. A utilização dessa ferramenta é apresentada no Guia de utilização do Gitlab (Anexo C), criado pela empresa.

A partir dessa ferramenta que são geridas e executadas todas as atividades de execução do desenvolvimento do projeto. Com o planejamento do Sprint realizado, o *Sprint Backlog* é fechado e a equipe assume a responsabilidade de sua realização.

A primeira tarefa é a de criação do protótipo, realizada pelo designer. Nesse momento, o designer passa a projetar a *interface* do sistema, buscando atender as *User Storys* e seus detalhamentos. Para isso, são utilizando técnicas de interação do usuário e experiência de usuário (UI/UX) e são definidas as características visuais e de usabilidade das funcionalidades.

Após feito o protótipo e aprovado pelo analista de negócio junto ao cliente, entra a etapa de implementação. É a etapa em que o sistema será efetivamente produzido. Os documentos e arquivos gerados nas etapas anteriores - *User Storys*, protótipo de tela e respectivos detalhamentos - são a base e suporte para os programadores entenderem o problema e consigam criar as soluções para o sistema.

Efetuada o desenvolvimento, os desenvolvedores geram uma versão do sistema, chamada *alpha*, que somente a equipe do projeto tem acesso. A geração de versão é orientada por um cronograma por Sprint, chamado *Template de Sprint* (Anexo E).

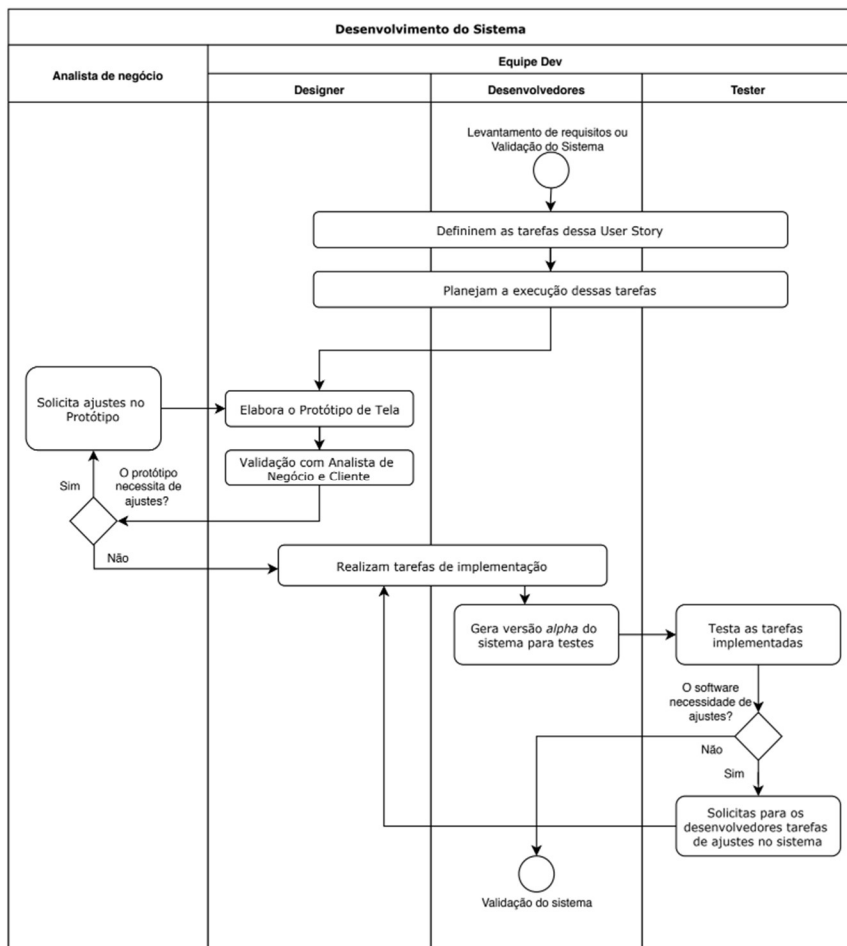


Figura 12 - Etapa de desenvolvimento do sistema.

Fonte: elaborado pelo autor.

O teste de software, próximo passo do processo, serve para encontrar possíveis erros que um programa recém-desenvolvido possa apresentar, de modo a conseguir corrigi-los antes que seja lançado no mercado, ficando disponível para o usuário final e/ou cliente.

Nas técnicas utilizadas é possível identificar princípios e ferramentas de Engenharia de Qualidade, como a utilização de *Checklists* e determinação de indicadores de qualidade. Além disso, no cotidiano o

colaborador que trabalha com testes está habituado a realizar uma bateria de testes de diferentes naturezas e propósitos.

Com testes realizados e funcionalidades desenvolvidas aprovadas, tem-se o incremento realizado.

É gerada uma versão *beta*, para homologação com o cliente em *Sprint Review*. O analista de negócio, apresentam o incremento do *Sprint*. As validações são ilustradas na Figura 13.

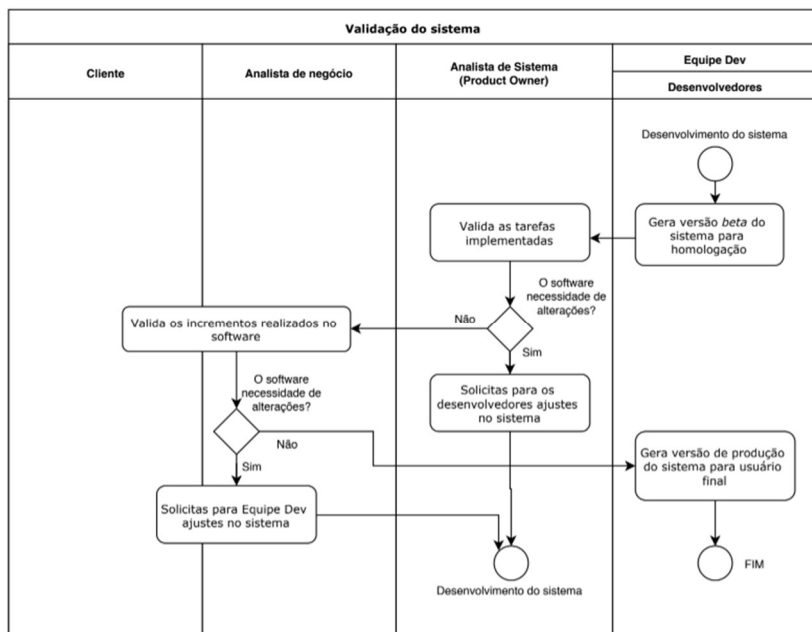


Figura 13 - Etapa de validação do sistema

Fonte: elaborado pelo autor.

As tarefas de alterações, correções e tarefas novas solicitadas nessa etapa de validação do sistema são entregues para equipe de desenvolvimento, para que sejam inseridas no planejamento do próximo *Sprint*. Todas as reuniões são registradas em Atas, para oficialização e posterior consulta (Anexo F e Anexo G).

4.3 AVALIAÇÃO DO PROCESSO

4.3.1 Considerações quanto a Gerenciamento de Projetos

- GPR 1. O escopo do trabalho para o projeto é definido;

Avaliação: Totalmente implementado.

Na etapa de iniciação dos projetos, o escopo do software é definido no Plano de trabalho (Anexo D), em uma visão de alto nível. Nele, é estabelecido o objetivo, motivação, os limites e restrições, os produtos, critérios de aceitação, entre outras informações.

- GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados, utilizando métodos apropriados;

Avaliação: Totalmente implementado.

No Plano de Trabalho (Anexo D), é possível também verificar a utilização de uma Estrutura Analítica do Projeto (EAP), onde a decomposição do trabalho apropriada é estabelecida. Nele, o escopo é dividido em etapas, que são divididos em atividades. Essas atividades serão utilizadas para gerar as *User Storys* que irão compor o *Product Backlog*, a lista de todo o trabalho a ser entregue no projeto.

- GPR 3. O modelo e as fases do ciclo de vida do projeto são definidos;

Avaliação: Totalmente implementado.

O modelo e suas fases são estabelecidos, já que a empresa utiliza de uma metodologia baseada no *Scrum*. Essa metodologia foi resumida em um *Template de Sprint* (Anexo E), em que são detalhados os pontos de controle e os marcos do *Sprint*.

- GPR 4. (Até 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;

Avaliação: Não implementado.

As estimativas de esforço e custo iniciais foram, nesse projeto, baseadas nos resultados dados históricos de projetos anteriores (Anexo D).

Já no ciclo de desenvolvimento, dentro do *Scrum*, o

dimensionamento não utiliza de nenhuma técnica. O dimensionamento é feito por mensuração aproximada de acordo com experiência dos integrantes. Assim, considera-se que o resultado esperado não é atendido.

- GPR 5. O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos;

Avaliação: Parcialmente implementado.

No Plano de Trabalho (Anexo D), também, são definidos os principais marcos, os objetivos de cada uma dessas entregas, a ordem certa para realização das atividades e os produtos liberados durante o projeto.

Vale ressaltar, que as atividades são divididas em níveis macro, o que dificulta as estimativas de atribuição de tamanho, esforço, cronograma e responsabilidades e é utilizada como uma estrutura subjacente para planejar, organizar e controlar o trabalho executado no projeto. Logo, temos isso como um ponto fraco do cronograma e, conseqüentemente, seu mantimento.

Já no desenvolvimento, o *Template de Sprint* (Anexo E) define o cronograma de atividades. Muitas vezes o *template* não é seguido, por diversos problemas relacionados ao desenvolvimento: impedimentos de desenvolvimento, estimativas mal realizadas, problemas na geração de versão, etc. Temos, portanto, outro ponto fraco no processo.

- GPR 6. Os riscos do projeto são identificados e seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;

Avaliação: Não implementado.

Não é feito um gerenciamento de riscos sistemático e estruturado. Os riscos do projeto são identificados de forma tácita pelos envolvidos no projeto e discutidos durante as reuniões de alinhamento, sem nenhuma técnica ou documentação estabelecida de levantamento e monitoramento desses riscos.

Por não estar documentado, o monitoramento, mesmo que tendo esforço, torna-se subjetivo e mal gerenciado.

Dessa forma, os riscos acabam sendo tratados de forma reativa. Nas *Daily Meetings* são identificados os riscos internos e atacados em tempo real, porém, não são documentados e monitorados.

- GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;

Avaliação: Totalmente implementado.

Quanto aos recursos humanos, o perfil e o conhecimento necessários para executá-lo foram considerados no planejamento. Esse planejamento é registrado no Plano de Trabalho (Anexo D), como parte do orçamento. Para o projeto em questão, foram realizadas no início do projeto, as contratações de seis dos oito colaboradores e os outros dois foram remanejados, para satisfazer as necessidades de mão-de-obra do projeto.

- GPR 8. (Até o nível F) Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;

Avaliação: Totalmente implementado.

Quanto aos recursos e ambiente em geral, são estabelecidos no início do projeto, no Plano de Trabalho (Anexo D), como base principalmente os recursos humanos, já que se trata de desenvolvimento de software e o principal recurso de produção é um computador para cada colaborador. Além disso, a necessidade de infraestrutura de servidores e ambientes de desenvolvimento também são estimadas a partir do Plano de Trabalho pelo *Scrum Master* e *Product Owner* junto ao administrativo da empresa.

- GPR 9. Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;

Avaliação: Totalmente implementado.

Para isso, a empresa oficializou algumas ferramentas de comunicação, registro e acompanhamento:

- E-mail: principal meio de comunicação entre diferentes projetos, os níveis da empresa e com o cliente.

- *Enterprise Architect*: uma plataforma de desenvolvimento colaborativa para modelagem, design e gerenciamento do produto. Uma solução para visualização, análise, modelagem e rastreamento do produto.

- Gitlab: é um gerenciador de repositório de software que realiza controle de versões distribuído, com suporte a uma Wiki, fornecendo também gerenciamento de tarefas, em nível operacional, e apoio a integração contínua e entregas contínuas de versões do software.

- Mattermost: Meio de comunicação oficial dentro do projeto é um serviço de bate-papo on-line de código aberto e instalada no servidor interno da empresa. Nele são comunicados avisos gerais, publicações e conversas em geral, relacionadas ao projeto. Além disso, essa ferramenta de comunicação possui integração com o Gitlab, permitindo vinculações com as tarefas do projeto.

- GPR 10. Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos;

Avaliação: Totalmente implementado.

Ao longo do projeto, enquanto as *Sprints* vão acontecendo e os planos específicos vão se consolidando, o Plano de Trabalho é atualizado e revisado, prazos reestabelecidos, com a diretoria da empresa e com o cliente, de acordo com a necessidade. Como evidência, observou-se o histórico de versões no *Enterprise Architect* e no Gitlab.

- GPR 11. A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados;

Avaliação: Totalmente implementado.

Para a empresa, o estudo de viabilidade é feito na elaboração do Plano de Trabalho, visando medir a capacidade de se concluir o projeto com êxito e se isto estará de acordo com os objetivos de negócio e portfólio da empresa. Esse estudo de viabilidade é atualizado e revisado ao longo do projeto, de acordo com a necessidade e acontecimentos. Caso for observado necessário, nas reuniões de *Sprint*, o *Scrum Master* tem a responsabilidade de solicitar mais recursos para o projeto.

- GPR 12. O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido;

Avaliação: Não implementado.

O Plano de Trabalho não foi revisado com todos os envolvidos no projeto, para levantamento de dúvidas e entendimento geral do projeto. Tanto é verdade que alguns dos entrevistados, envolvidos diretamente no desenvolvimento do trabalho, não receberam, nem leram, o Plano de Trabalho.

A equipe está informada, geralmente, apenas em entregas imediatas, *Sprint* por *Sprint*.

Dessa forma, a visão do panorama geral do projeto é prejudicada, apenas entendendo sobre o projeto e produto o que lhes é requisitado e conversado em reuniões.

- GPR 13. O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado;

Avaliação: Totalmente implementado.

O monitoramento é realizado, em nível operacional e tático, nos *Daily Meetings* e em reuniões de *Sprints Reviews*.

Em níveis estratégico e para acompanhamento do cliente, são realizadas reuniões de *feedback* com os analistas de negócio e de sistemas, para repassar o andamento das atividades, retomada decisões, realização de ajustes no produto e repriorizações. Essas reuniões são registradas em Ata (Anexo F e Anexo G).

- GPR 14. Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado;

Avaliação: Totalmente implementado.

No projeto, esse controle foi realizado pelas ferramentas *Enterprise Architect* e *Gitlab*. Essas ferramentas fornecem históricos de versão de documentação e tarefas. Além disso, as tarefas podem ser designadas para os integrantes dos projetos, vinculadas a uma estimativa e tempo real de execução da tarefa e, conseqüentemente, do *Sprint*.

O evento *Sprint Retrospective* também é um forte aliado nesse monitoramento. Nessa reunião, o *Scrum Master* tem uma atualização da situação do projeto, para realizar esse monitoramento.

- GPR 15. Os riscos são monitorados em relação ao planejado;

Avaliação: Não implementado.

Como explicitado no GPR 6, os riscos não são documentados e acabam sendo tratados de forma reativa. Por não estar documentado, o monitoramento, mesmo que tendo esforço, torna-se subjetivo e mal gerenciado. Os riscos, em geral, são identificados nas reuniões diárias, revisão de *sprint* e retrospectiva da *sprint*.

- GPR 16. O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido;

Avaliação: Totalmente implementado.

No projeto, durante o planejamento do *Sprint*, as tarefas são designadas para os integrantes dos projetos no Gitlab. Os papéis de *Product Owner* e de *Scrum Master* são responsáveis por monitorar o envolvimento das partes interessadas. O Gitlab fornece esse monitoramento em uma de suas funcionalidades, o *Board*, uma ferramenta visual de acompanhamento de tarefas.

- GPR 17. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;

Avaliação: Totalmente implementado.

A revisão do Plano do Projeto (Anexo D) acontece em reuniões de gestão com os responsáveis pelo produto, em nível estratégico e com o cliente.

Após essas reuniões, são acordadas as alterações no plano do projeto e há um comprometimento de cumpri-los. Essas alterações muitas vezes alteram aspectos do escopo e altera prioridades e o próprio cronograma. O definido nessas reuniões é registrado em ata.

Já no nível operacional, a revisão e ajustes dos planos de projeto são realizados nos Eventos do *Scrum*. Com os feedbacks do *Sprint*

Review, é definido um plano no *Sprint Planning* para a próxima *Sprint* e assumido o compromisso da Equipe Dev em cumpri-lo.

Apesar desse compromisso, há a dificuldade em mantê-lo, já que as estimativas das tarefas são realizadas de forma tácita, sem um método estabelecido.

De forma resumida, pergunta-se para cada um quantas horas ele supõe que irá demorar para realizar aquela tarefa.

Quanto aos marcos do projeto, ao fim de cada *Sprint* há pontos de monitoramento de revisão chamado *Sprint Review*, que proporciona monitoramento em nível micro. Em marcos do Plano de Trabalho, realizado após um número definido de *Sprints*, são realizadas revisões do produto. Observando que, esses marcos podem ser mudados de acordo com a necessidade do desenvolvimento e resultante de alterações de demanda do cliente.

- GPR 18. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;

Avaliação: Totalmente implementado.

Após o monitoramento, realizado no *Sprint Review*, *Sprint Retrospective*, são listados os problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas. Isso é registrado em Ata (Anexo G).

- GPR 19. Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão;

Avaliação: Totalmente implementado.

Quanto a corrigir desvios em relação ao planejado, cabe ao *Scrum Master* realizar tarefas para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas. Em geral, problemas urgentes são tratados imediatamente ao serem identificados. Já outros problemas, são considerados como tarefas novas a serem tratadas no próximo *Sprint*. Isso é registrado em Ata (Anexo G).

4.3.2 Considerações quanto a Gerenciamento de Requisitos

- GRE 1. O entendimento dos requisitos é obtido junto aos fornecedores de requisitos;

Avaliação: Totalmente implementado.

No projeto, a principal forma de comunicação é o e-mail e reuniões presenciais, registradas formalmente em Atas (Anexo F). Além disso, muitas vezes são coletados documentos, planilhas, e outros artefatos que o cliente usa relacionado com o projeto. Essas informações são úteis para gerar o entendimento da necessidade do cliente e os requisitos para satisfazê-las.

- GRE 2. Os requisitos são avaliados com base em critérios objetivos e um comprometimento da equipe técnica com estes requisitos é obtido;

Avaliação: Parcialmente implementado.

O escopo do trabalho é dividido em produtos, que são divididos em atividades, no Plano de trabalho. O entendimento dos requisitos é obtido junto aos fornecedores de requisitos, porém os requisitos são avaliados com base em critérios subjetivos.

Todas as etapas e tarefas estão definidas na visão de negócio e do cliente. A avaliação dos requisitos deve envolver, além do cliente, também, a equipe técnica da organização. É considerado um ponto fraco do processo, já que os critérios precisam ser mais objetivos.

- GRE 3. A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;

Avaliação: Totalmente implementado.

Essa rastreabilidade é realizada no *Enterprise Architect*. De modo que, é possível vincular os requisitos entre si, às *User Storys* e ao produto. Além disso, as tarefas do Gitlab são vinculadas às *User Storys* que as originaram. Dessa forma, essa rastreabilidade ajuda na avaliação do impacto nas alterações de requisitos ao longo do ciclo de vida do projeto.

- GRE 4. Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;

Avaliação: Totalmente implementado.

Revisões em planos e produtos de trabalho do projeto é realizado pelo PO, durante principalmente as *Sprint Reviews*. Além disso, o *tester* tem a função de fazer esse paralelo diário entre o que foi solicitado em documentação e o que foi entregue em sistema, incluindo o funcionamento adequado do desenvolvido. Ademais, a ferramenta utilizada no desenvolvimento fornece o encadeamento dessas tarefas nas suas respectivas *User Storys*, *Epics* e *Themes*.

Caso os *testers*, em suas tarefas diárias e o analista de sistemas, no *Sprint Review* verifique inconsistências entre o documentado e o desenvolvido, é realizado uma tarefa de correção da documentação ou do produto, dependendo da situação.

- GRE 5. Mudanças nos requisitos são gerenciadas ao longo do projeto.

Avaliação: Totalmente implementado.

Esse versionamento de requisitos é realizado no *Enterprise Architect*, de modo possível de rastrear todas as mudanças realizadas nos requisitos, contendo um histórico de alterações, permitindo seu gerenciamento.

4.3.3 A avaliação quanto ao nível G do MR-MPS-SW

Com essas considerações, pode-se realizar a avaliação do processo do projeto. Dentre os 24 resultados de processo avaliados, 18 foram considerados totalmente implantados. Dos 6 restantes:

- Largamente Implementados (L): GRE 2
- Parcialmente implementados (P): GPR 5
- Não Implementados (N): GPR 4, GPR 6, GPR 12 e GPR 15

A Tabela 6, resume esses resultados, destacando as GPR e GRE com problemas identificados.

Tabela 6 - Caracterização do grau de implementação de cada resultado esperado.

Processo e Atributos de processo do modelo do Modelo	Há indicador direto adequado?	Há uma afirmação confirmando implementação?	Encontrado ponto fraco substancial?	Avaliação
GPR 1	SIM	SIM	NÃO	T
GPR 2	SIM	SIM	NÃO	T
GPR 3	SIM	SIM	NÃO	T
GPR 4	SIM	NÃO	NÃO	N
GPR 5	SIM	SIM	SIM	P
GPR 6	NÃO	NÃO	SIM	N
GPR 7	SIM	SIM	NÃO	T
GPR 8	SIM	SIM	NÃO	T
GPR 9	SIM	SIM	NÃO	T
GPR 10	SIM	SIM	NÃO	T
GPR 11	SIM	SIM	NÃO	T
GPR 12	NÃO	NÃO	SIM	N
GPR 13	SIM	SIM	NÃO	T
GPR 14	SIM	SIM	NÃO	T
GPR 15	NÃO	SIM	SIM	N
GPR 16	SIM	SIM	NÃO	T
GPR 17	SIM	SIM	NÃO	T
GPR 18	SIM	SIM	NÃO	T
GPR 19	SIM	SIM	NÃO	T
GRE 1	SIM	SIM	NÃO	T
GRE 2	SIM	SIM	SIM	L
GRE 3	SIM	SIM	NÃO	T
GRE 4	SIM	SIM	NÃO	T
GRE 5	SIM	SIM	NÃO	T

Fonte: elaborado pelo autor. Legenda: (T) totalmente implementado, (L)

largamente implementado, (P) Parcialmente implementado, (N) Não implementado.

Quanto aos atributos de processo e seus RAPs, os resultados da avaliação podem ser verificados no Apêndice B. Um resumo dessa caracterização do grau de implementação de cada RAP e AP é apresentado na Tabela 7.

Tabela 7 - Caracterização do grau de implementação de cada RAP e AP.

Atributo de Processo		Porcentagem	Avaliação do RAP	Avaliação do AP
AP1.1	(i)	0,792	L	L
AP 2.1	(i)	0,875	T	T
	(ii)	0,792	T	
	(iii)	0,792	T	
	(iv)	0,792	T	
	(v)	0,792	T	

Fonte: elaborado pelo autor. Legenda: (T) totalmente implementado, (L) largamente implementado, (P) Parcialmente implementado, (N) Não implementado.

Após caracterizar o grau de implementação de cada resultado de processo e de cada atributo de processo na unidade organizacional, foi caracterizado o grau de implementação do processo como um todo:

NÃO SATISFEITO.

Utilizando como parâmetro o nível G do MR-MPS-SW, todos os resultados esperados para o processo deveriam ser caracterizados como T (Totalmente implementado) ou L (Largamente implementado). No entanto, cinco resultados esperados de processos (GPR 4, GPR 5, GPR 6, GPR 12 e GPR 15) não atingiram o mínimo grau de implantação necessário estabelecido no modelo para satisfação do Nível G.

Quanto aos Atributos de Processo, para o AP1.1 deve ser totalmente implementado e nenhum de dos processos pode ser N (Não implementado). No entanto, ficou classificado como largamente implementado e cinco processos reprovaram nesse atributo.

Quanto ao AP 2.1, como seus RAP foram classificados em T (Totalmente implementado), de acordo com o modelo de avaliação, o AP está implementado.

Tendo em vista esses resultados, faz-se necessário apresentar propostas de melhorias nos problemas encontrados.

4.4 PROPOSTAS DE MELHORIA

Essa seção tratou de propostas para melhoramento desse processo para que alcance os Resultados Esperados. As propostas serão realizadas para todos os Resultados Esperados de Processo que não alcançaram a caracterização de Totalmente implementado.

4.4.1 GPR4

O GPR 4 trata do dimensionamento de esforço e custo para a execução das tarefas e dos produtos de trabalho. O problema identificado é que essa estimativa não é feita com nenhuma técnica adequada. Dessa forma, o dimensionamento proposto é utilizando uma técnica chamada *Planning Poker*.

De acordo com Rubin (2012), o *Planning Poker* é uma técnica baseada em consenso para estimar o esforço. Nele, os integrantes se envolvem em uma intensa discussão para expor suas opiniões, adquirir um entendimento compartilhado e dimensionar as tarefas.

O *Planning Poker* gera estimativas de tamanho relativo, agrupando com precisão itens de tamanho semelhante. A equipe aproveita o histórico de estimativas de tarefas estabelecido para estimar com mais facilidade o próximo conjunto de tarefas. (RUBIN, 2012).

A escala mais frequentemente usada é baseada em uma sequência de Fibonacci modificada: 1, 2, 3, 5, 8, 13, 20, 40 e 100. Essas escalas estão em um baralho de cartas (Figura 14), que é entregue para cada um dos membros da Equipe Dev. (RUBIN, 2012).

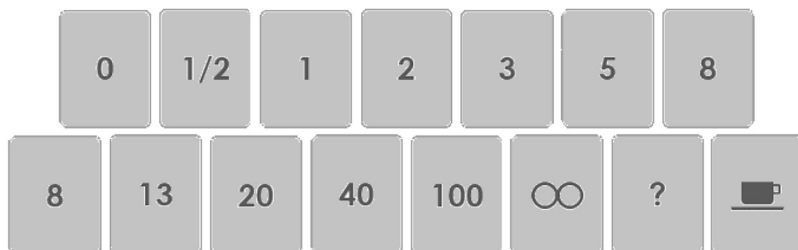


Figura 14 - Cartas de Planning Poker.
Fonte: Visual Paradigm (2019).

As regras para execução do Planning Poker são as seguintes (RUBIN, 2012):

1. O PO seleciona uma tarefa a ser estimada e lê para a equipe.
2. Os membros da equipe de desenvolvimento discutem o item e fazem perguntas esclarecedoras ao PO, que responde às perguntas.
3. Cada estimador seleciona secretamente um cartão que representa sua estimativa.
4. Depois que cada estimador faz uma seleção privada, todas as estimativas são expostas simultaneamente a todos.
5. Se todos selecionarem o mesmo cartão, tem-se consenso e esse número de consenso se torna a estimativa da tarefa.
6. Se as estimativas não forem as mesmas, os membros da equipe se envolvem em uma discussão focada para expor suposições e mal-entendidos. Normalmente, começa-se pedindo aos estimadores mais alto e mais baixo que expliquem ou justifiquem suas estimativas.
7. Após a discussão, retornamos à etapa 3 e repete-se até chegar a um consenso.

A vantagens da utilização da técnica são citadas por Rubin (2012): as estimativas em grupo são geralmente mais precisas que estimativas individuais e a discussão gera melhor entendimento da tarefa pelos participantes da equipe.

4.4.2 GPR5 e GRE2

O GPR5 trata se o orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos. No entanto, como ponto fraco encontrado nesse GPR, identificou-se que os pontos de controle e marcos dificilmente são

mantidos. Esse problema resulta principalmente de problemas relacionados a relação estimativa/execução.

Uma das causas da execução não realizar o planejado é a má estimativa das atividades de desenvolvimento ainda na etapa de planejamento. Por sua vez, o GRE2 destaca a necessidade de realizar essa estimativa com base em critérios objetivos e obtenção do comprometimento da equipe técnica com esses requisitos e cronogramas definidos. Atualmente, as estimativas do Plano de trabalho são avaliadas a partir das atividades, que são subjetivas e de alto nível.

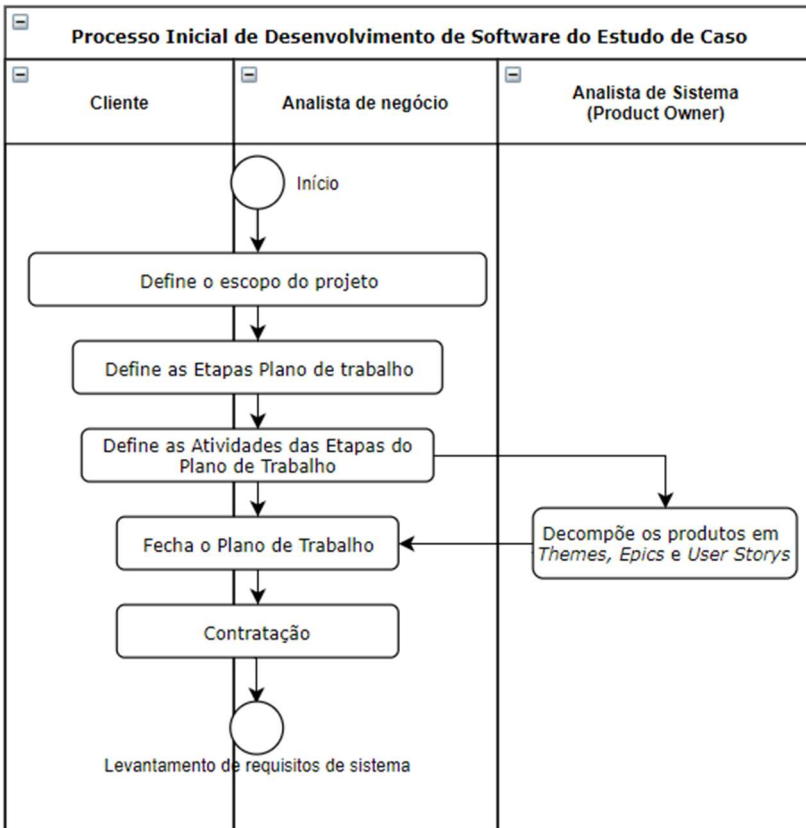


Figura 15 - Processo inicial com tarefa de decomposição em *User Storys*.

Fonte: elaborado pelo autor.

Para resolver esses dois problemas, propõe-se que seja inserida uma reunião com a equipe técnica para avaliar as atividades ainda nas

etapas de iniciação do projeto, antes de fechamento do cronograma e marcos. Dessa forma, o detalhamento de produtos deve ser realizado em *User Storys*, que são mais objetivos que as Atividades do Plano de Trabalho e fornecem um dimensionamento melhor e mais objetivo do produto, podendo fazer parte da elaboração do Plano de trabalho.

Esse detalhamento, que era realizado posteriormente da contratação do serviço, seria adiantado, como apresentado na Figura 15.

Outro problema relacionado a manter o cronograma está relacionado com a execução. Para apoiar o monitoramento da execução desses marcos e pontos de controle, além dos Eventos *Scrum*, um Gráfico de Evolução Regressiva de Iteração (Gráfico *Burndown*) pode ser utilizado para prever o progresso e adiantar as ações corretivas.

No Guia PMBOK (PMI, 2017, p. 226) é definido:

“Este gráfico rastreia o trabalho que ainda precisa ser concluído na lista de pendências de iteração. É usado para analisar a variação em relação a uma evolução regressiva ideal com base no trabalho comprometido desde o planejamento da iteração. Uma linha de tendência de previsão pode ser usada para prever a provável variação na conclusão da iteração e adotar medidas apropriadas durante o curso da iteração. Uma linha diagonal que representa a evolução regressiva ideal e o trabalho restante real diário é então desenhada. Uma linha de tendência é então calculada para prever a conclusão com base no trabalho restante.”

Esse gráfico, portanto, tem a capacidade de aumentar a possibilidade do cronograma, marcos e pontos de controles serem mantidos. A Figura 16 exemplifica esse gráfico.

Nas entrevistas, foi informado que o Gráfico *Burndown* não é utilizado no projeto, pois as estimativas não estavam bem alinhadas com a realidade. Com chances pequenas de alcançar essas metas estabelecidas previamente, a equipe perdeu interesse na ferramenta. Com estimativas melhores no planejamento, melhor controle pode ser realizado e os marcos e pontos de controle terão maiores chances de serem mantido.



Figura 16 - Gráfico de Evolução Regressiva de Iteração (*Burndown*).
Fonte: PMI (2017).

Vale observar que, por tratar-se de um sistema customizado, feito sob medida para o cliente, a mudança de requisitos é mais comum, o que também prejudica o cumprimento das estimativas planejadas. Porém, essas estimativas deverão ser refeitas sempre que houver uma mudança e um novo cronograma estabelecido.

4.4.3 GPR6 e GPR15

Riscos são inerentes a todo e qualquer projeto não só os oriundos do próprio projeto, mas também os relacionados a contratante (data e qualidade das especificações, por exemplo). O diferencial esperado nesses resultados é o registro e monitoramento dos riscos, além da probabilidade que ocorram e a respectiva prioridade para tratamento. (SOFTEX, 2016).

O gerenciamento dos riscos do projeto inclui os processos de condução do planejamento, da identificação, da análise, do planejamento das respostas, da implementação das respostas e do monitoramento dos riscos em um projeto. (PMI, 2017)

Segundo a PMI (2017), o gerenciamento dos riscos do projeto tem por objetivo aumentar a probabilidade e/ou o impacto dos riscos positivos e diminuir a probabilidade e/ou o impacto dos riscos negativos, a fim de otimizar as chances de sucesso do projeto.

No plano de gerenciamento dos riscos do Guia PMBOK consta que é importante manter uma planilha de riscos, contendo dados do risco,

descrição, probabilidade, impacto e prioridades no seu tratamento, para acompanhamento de como afetam o projeto e para se tomar ações. Uma ferramenta que apoia essa análise é a Matriz SWOT. (PMI, 2017)

Essa ferramenta é a Análise de forças, fraquezas, oportunidades e ameaças, conhecido como Matriz SWOT (*Strengths, Weaknesses, Opportunities and Threats*). Esta técnica examina o projeto com base em cada uma destas perspectivas. (PMI, 2017).

Além do mais, na identificação dos riscos, é utilizada para aumentar a amplitude dos riscos identificados incluindo riscos gerados internamente. A técnica começa com a identificação das forças e fraquezas da organização, com foco no projeto, na organização ou na área do negócio em geral. Em seguida, a análise SWOT identifica as oportunidades do projeto resultantes das forças da organização, assim como as ameaças decorrentes das fraquezas. A análise também examina o grau com que as forças da organização podem compensar as ameaças e determina se as fraquezas poderiam impedir as oportunidades. (PMI, 2017).

A Figura 17, apresenta a Matriz SWOT.

	AJUDA	ATRAPALHA
INTERNA (organização)	FORÇA	FRAQUEZA
EXTERNA (ambiente)	OPORTUNIDADES	AMEAÇAS

Figura 17 - Matriz SWOT.

Fonte: adaptado de Rocha (2016).

Martins (2007) descreve e definir os componentes da matriz SWOT:

- Forças: são as características positivas do projeto relacionadas ao produto que é oferecido. Deve conter aspectos controlados pela equipe e relevantes para a estratégia do projeto;
- Fraquezas: são as características negativas do projeto relacionadas ao produto que é oferecido. Deve conter aspectos controlados pela equipe e relevantes para a estratégia do projeto;
- Oportunidades: são as características positivas do produto em relação ao mercado. Deve conter aspectos que não podem ser controlados pela empresa e relevantes para a estratégia do projeto;
- Ameaças: são as características negativas do produto em relação ao mercado. Deve conter aspectos que não podem ser controlados pela empresa e relevantes para a estratégia do projeto;

Logo, cabe ao líder da equipe, no caso o *Scrum Master* e o *Product Owner* o gerenciamento dos riscos. A coleta dos dados relacionados ao risco deve ser realizada nos eventos *Scrum* e seu monitoramento tratados também durante as *Sprints*.

4.4.4 GPR 12

Como o problema relacionado ao GPR12 está na desinformação da equipe com relação a visão macro do projeto e entendimento de suas tarefas, propõe-se a inserção de uma reunião de *kick-off* logo após a contratação do serviço. (PMI, 2017).

De acordo com Wysocki (2014, pag.):

“A Reunião de *kick-off* é o anúncio formal à organização de que este projeto foi planejado e aprovado para execução. Essa reunião acontece apenas uma vez em cada projeto - no início do projeto, após a aprovação do plano e do projeto, mas antes da conclusão de qualquer trabalho. Não é apenas uma reunião para os membros da equipe, mas também é sua oportunidade de começar o projeto com um bom começo.”

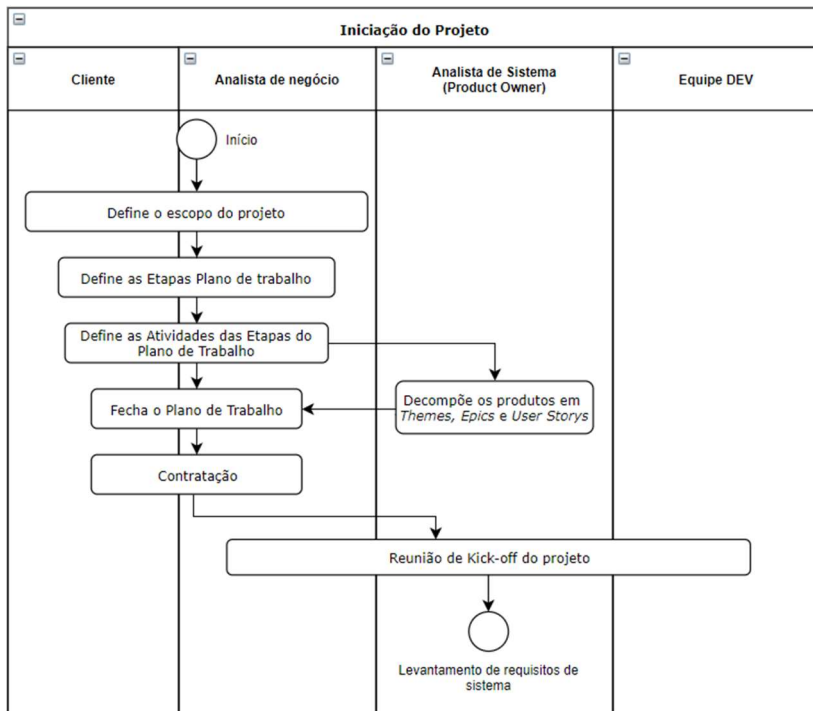


Figura 18 - Processo de iniciação com as alterações sugeridas para atender o GPR12.

Fonte: elaborado pelo autor.

Pode ser utilizada para melhorar o entendimento e obter o comprometimento tanto dos participantes internos do projeto quanto externos. A Figura 18 mostra como ficaria o processo, com a reunião de *kick-off* inserida ao fim da etapa de iniciação do projeto.

Essa reunião fornece um pontapé inicial eficaz, tanto internamente quanto na reunião inicial do cliente, para iniciar o projeto com o pé direito, seguindo na direção certa, o mais rápido possível.

Vale ressaltar que, ao término de cada *Sprint*, o plano do projeto deve ser revisado junto ao time reafirmando o comprometimento.

4.4.5 5W2H das propostas de melhoria

Para expor com clareza essas propostas de melhorias sugeridas, foi utilizado a ferramenta 5W2H para organizá-las. Essa Tabela foi inserida nesse trabalho na Figura 19 e na Figura 20.

5W				2H		
O que? O que deve ser feito?	Por que? Porque deve ser implementado?	Quem? Quem é o responsável pela ação?	Onde? Onde deve ser executado?	Quando? Quando deve ser implementado?	Como? Como deve ser conduzido?	Quanto? Quanto vai custar a implementação?
Estimar tarefas utilizando uma técnica adequada	Para satisfazer o resultado esperado do GPR4, que demanda uma técnica adequada no dimensionamento das tarefas.	Toda a Equipe Scrum, sendo coordenado pelo <i>Product Owner</i>	Reunião de <i>Sprint Planning</i>	A cada <i>Sprint</i> (de duas em duas semanas)	Utilizando as cartas de <i>Planning poker</i> e seguindo as 7 regras para execução do <i>Planning Poker</i> .	Na Amazon, o preço das cartas (para 8 jogadores) está aprox. 40 reais. Além disso, será demandado mais tempo de <i>Sprint Planning</i> .
Realizar reunião com equipe técnica para fechamento do cronograma	Para satisfazer o GPR5 e GRE2, com objetivo de oferecer critérios objetivos para estabelecimento do organograma e cronograma.	Analista de Negócio e uma equipe técnica composta por analistas de sistemas e desenvolvedores experientes.	Na sala de reunião da empresa e por vídeo conferência, se necessário.	Após a determinação das etapas de projeto e suas atividades e antes do estabelecimento do cronograma e organograma.	Em uma reunião, onde as atividades do plano de trabalho são quebradas até o nível de <i>User Storys</i> e tempo e recursos necessários para cumpri-las são estimados pela equipe técnica.	Uma única reunião de uma tarde com o cliente, tomando o tempo dos seus integrantes nesse dia.
Acompanhamento da evolução das <i>User Storys</i> e tarefas.	Satisfazer o GPR5, aumentando a probabilidade do cumprimento do cronograma	O <i>Scrum Master</i> deverá manter e atualizar o gráfico para apresentar a equipe.	No <i>Daily Meeting</i> para acompanhar as tarefas e no <i>Sprint Review</i> para acompanhar as <i>User Storys</i>	Diariamente para tarefas e todo <i>Sprint</i> para <i>User Storys</i>	Utilizando Gráficos <i>Burn-down</i>	Um pequeno tempo do <i>Scrum Master</i> para atualizar o gráfico. .

Figura 19 - Análise 5W2H (parte 1/2).

Fonte: elaborado pelo autor.

5W			2H			
O que? O que deve ser feito?	Por que? Porque deve ser implementado?	Quem? Quem é o responsável pela ação?	Onde? Onde deve ser executado?	Quando? Quando deve ser implementado?	Como? Como deve ser conduzido?	Quanto? Quanto vai custar a implementação?
Gerenciar os Riscos do projeto	Para satisfazer o GPR 6 e GPR15, que tratam de definição e monitoramento de riscos do projeto	PO para oportunidades e ameaças (externos) e Scrum Master para forjas e fraquezas (interno).	Em uma planilha compartilhada entre os PO e o SM.	Diariamente, mais intensamente no <i>Sprint Review</i> e <i>Sprint Retrospective</i>	Utilizando uma planilha que deve ser acompanhada, atualizada e ações devem ser tomadas, quando necessário.	Um pequeno tempo do SM e do PO para atualizar a planilha.
Fornecer um panorama geral do projeto para todos os interessados	Para atender o GPR12, que demanda a revisão do Plano de Projeto com todos os interessados	Todos os envolvidos no projeto.	Na sala de reunião da empresa e por vídeo conferência, se necessário.	Logo após o fechamento do Plano de Trabalho e a contratação do projeto	Essa reunião acontece apenas uma vez em cada projeto, onde é anunciado o projeto e apreendido o que foi planejado para execução	Uma única reunião de uma tarde com o cliente, tomando o tempo dos seus integrantes do projeto nesse dia.

Figura 20 - Análise 5W2H (parte 2/2).

Fonte: elaborado pelo autor.

Como pode-se observar, essa ferramenta ajudou a identificar e responder os principais questionamentos relacionados a futura implantação das práticas sugeridas. Nessa situação, que envolvem a implementação de várias decisões de forma orquestrada, o 5W2H se mostra bastante eficaz.

5 CONCLUSÕES

As Tecnologias de Informação (TI) são elementos propulsores do novo padrão mundial onde conhecimento e informação tem papéis centrais no mercado.

As empresas fornecedoras de sistemas de informação deparam-se com alguns desafios cotidianos relacionados à qualidade, que essas metodologias não atendiam. Para isso, é comum investir em qualidade através do melhoramento dos processos.

Ao investigar as formas de melhorias de processos de software utilizadas pelas empresas brasileiras, um dos objetivos dessa pesquisa, descobriu-se que, dentre as formas de buscar avanços na qualidade do processo, basear-se em modelos de referência e de maturidade tem se mostrado um caminho bastante utilizado pelas empresas.

Ademais, um dos métodos se destacou: o MPS.BR, por ser uma alternativa criada especialmente focada no cenário nacional, focada em empresas de porte menor e menos onerosa em comparação com as alternativas internacionais.

Assim, o segundo objetivo específico do trabalho pode ser estabelecido: mapear os processos necessários para a implantação do MPS.BR. Ao compreender como o modelo funciona, definiu-se o Nível G, nível inicial do modelo, como ponto de partida.

Com isso em mente, procurou-se identificar um caso a ser estudado para implantação desse modelo: um projeto de desenvolvimento de um repositório de conteúdo para educação de trânsito.

Baseando-se no Método de Avaliação (MA-MPS), foi possível então, mapear o processo de desenvolvimento desse repositório. Com esse mapeamento, obteve-se sucesso em levantar a situação atual do processo de produção de software do projeto.

Esse mesmo método de avaliação permitiu que fosse verificado a conformidade desse processo com o nível G do Modelo de Referência MPS para Software.

Como resultado dessa atividade, que foi também objetivos específicos desse trabalho, descobriu-se algumas inconsistências do processo estudado com o modelo, que causam o processo ser classificado

como “Não satisfeito” na avaliação:

1. O esforço e o custo para a execução das tarefas não eram estimados com base em referências técnicas, eram feitos de forma intuitiva;
2. O orçamento e cronogramas eram estabelecidos com base em fontes subjetivas e dificilmente eram cumpridos;
3. Os riscos não eram devidamente identificados, documentados e monitorados;
4. O Plano do Projeto não era revisado com todos os interessados;
5. Os requisitos não eram avaliados com base em critérios objetivos;

Essas inconsistências foram refletidas na avaliação dos processos GRE 2, GPR 4, GPR 5 GPR 6, GPR 12 e GPR 15 e em seus atributos de processo.

Com essas inconsistências mapeadas, foi possível identificar e propor melhorias de procedimentos que agregam valor aos produtos desenvolvidos com foco nos clientes. Essas melhorias foram levantadas utilizando a ferramenta 5W2H e detalhadas ao final do estudo de caso. Sendo elas:

1. Estimar as tarefas de desenvolvimento utilizando a técnica *Planning Poker*;
2. Mover a etapa de decomposição dos produtos e atividades em *Themes*, *Epics* e *User Storys* para a iniciação do projeto, inserindo no Plano de Trabalho;
3. Utilizar a ferramenta Gráfico *Burndown* para acompanhamento das Sprints e realização das tarefas;
4. Utilizar uma planilha compartilhada com uma Matriz SWOT para gerenciamento de riscos do projeto;
5. Realizar uma reunião de *kick-off* com os envolvidos do projeto para apresentar o planejado para o projeto e fornecer um panorama geral de sua execução;

Aplicar essas propostas permitirá que a empresa alcance o nível G do MPS.BR e que a melhoria de processos comece a se tornar parte integrante do desenvolvimento organizacional e cultural na empresa.

Recomenda-se que futuramente a empresa realize uma reavaliação para verificar os avanços resultantes das ações de implantações

realizadas. Com isso, a empresa pode criar o hábito de realizar medições periódicas (anual ou semestral, por exemplo) para monitorar seu processo como um todo e realizar os ajustes necessários.

Vale ressaltar que cabe a empresa aprofundar-se nas melhorias propostas ou encontrar alternativas próprias para o alcance dos resultados esperados do nível G do MPS.BR e as vantagens associadas a isso. O escopo deste trabalho limitou-se a avaliação e elaboração de propostas.

Destaca-se, também, que o nível G é apenas o primeiro nível do MPS.BR. Como esse modelo é acumulativo, sugere-se avançar na avaliação para os níveis superiores quando alcançados os níveis inferiores. Dessa forma, caso as implantações sugeridas ou equivalentes sejam bem-sucedidas e o nível G alcançado, a empresa pode investir em buscar o nível F e assim sucessivamente.

Pode-se, também, utilizar das vantagens dessa medição em outros projetos e evoluir a organização como um todo. Os avanços positivos no projeto piloto deste estudo de caso, podem ser institucionalizados.

Sugere-se, também, que as vantagens relacionadas a certificação sejam avaliadas e, caso estejam alinhadas aos objetivos da empresa, seja realizada uma certificação junto ao órgão avaliador do MPS.BR.

REFERÊNCIAS BIBLIOGRÁFICAS

ABES - Associação Brasileira das Empresas de Software. **Mercado Brasileiro de Software: panorama e tendências**, 2019. Disponível em: <<http://central.abessoftware.com.br/Content/UploadedFiles/Arquivos/Dados%202011/ABES-EstudoMercadoBrasileirodeSoftware2019.pdf>>. Acesso em: agosto de 2019.

AHERN, D. M., CLOUSE, A. et al. **CMMI distilled**. Addison-Wesley. 2001.

ALBUQUERQUE, A. **Avaliação e Melhoria de Ativos de Processos Organizacionais em Ambientes de Desenvolvimento de Software** - Tese de Doutorado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008.

ALVES, Thiago Salhab. **MAPSW: UM MÉTODO DE AUTO-AVALIAÇÃO DE PROCESSOS DE SOFTWARE FRENTE AO CMMI NÍVEL 2**. Albuquerque Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Metodista de Piracicaba, Piracicaba, 2008.

BECK, K. et al. **Manifesto para o desenvolvimento ágil de software**, 2001. Disponível em: <<https://www.manifestoagil.com.br/index.html>>. Acesso em: agosto de 2019.

CAVALCANTI, Ana Regina R.; SANTOS Gleison S.; PERINI Monalessa B. **MEDIÇÃO DE SOFTWARE e Controle Estatístico de Processos**. Brasília, 2012.

CMMI INSTITUTE. **What is CMMI?** Disponível em: <<https://cmmiinstitute.com/cmmi/intro>>. Acesso em: agosto de 2019.

COUTINHO, Thiago. **O que é o ciclo PDCA?** Disponível em: <<https://www.voitto.com.br/blog/artigo/o-que-e-o-ciclo-pdca>>. Acesso em: outubro de 2019.

FALBO, R. A. **Integração de Conhecimento em um Ambiente de Desenvolvimento de Software**. Tese (Doutorado em Ciências em Engenharia de Sistemas e Computação), COPPE/UFRJ, Rio de Janeiro, 1998.

FALBO, R.; BERTOLLO, G., **Establishing a Common Vocabulary for Helping Organizations to Understand Software Processes, Proceedings of the 1st Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE)**, Enschede, The Netherlands, ISSN 1574-0846, 2005.

FELIZ, Tom. **Lightweight Software Process Assessment and Improvement**, 2012. Disponível em: <http://www.uploads.pnsgc.org/2012/papers/t-57_Feliz_paper.pdf>. Acesso em: agosto de 2019.

FUMSOFT. **Pesquisa revela melhoria nas empresas que adotaram MPS.Br**, 2011. Disponível em: <<http://www.fumsoft.org.br/noticias/pesquisa-revela-melhoria-nas-empresas-que-adotaram-mps-br>>. Acesso em: agosto de 2019.

GONÇALO, C. R., Junges, F. M. e Borges, M. L. **Avaliação da gestão do conhecimento: Modelos de mensuração**, XXX Encontro Nacional de Engenharia de Produção, (2010).

HAUCK, J. C. R. and von Wangenheim, C. G. **A method for software process capability/maturity models customization to specic domains, Software Engineering (SBES)**, 2011 25th Brazilian Symposium on, IEEE, 2009, pp. 293–302.

HUMPHREY, W. **Managing the software process**. Massachussets: Addison-Wesley, 1989.

IBGE, **O Setor de Tecnologia da Informação e Comunicação no Brasil | 2003-2006**, 2009. Disponível em: <<https://www.ibge.gov.br/estatisticas/multidominio/ciencia-tecnologia-e-inovacao/9139-o-setor-de-tecnologia-da-informacao-e-comunicacao-no-brasil.html?=&t=o-que-e>>. Acesso em: agosto de 2019.

ISO/IEC - INTERNATIONAL ORGANIZATION FOR STANDARDIZATION/ INTERNATIONAL ELECTROTECHNICAL COMISSION. **ISO/IEC 12207 Systems and software engineering–Software life cycle processes**. Geneve: ISO, 2008.

_____. **ISO/IEC 33020:2015: Information Technology - Process**

Assessment – Process measurement framework for assessment of process capability. Geneve: ISO, 2015.

KALINOWSKI, M., Santos, G., Reinehr, S., Montoni, M., Rocha, A. R., Weber, K. C. and Travassos, G. H. **Mps. br: promovendo a adoção de boas práticas de engenharia de software pela indústria brasileira**, XIII Congresso Iberoamericano em "Software Engineering" (CIBSE). Cuenca: Equador, (2010).

LIMA, Aline Francielle Dos Anjos. **Processo de desenvolvimento de requisitos do CMMI-DEV e MR-MPS-SW para projetos de manutenção evolutiva: um estudo de caso de implementação.** Dissertação (Mestrado) - Universidade Federal do Pará, Instituto de Ciências Exatas e Naturais, Programa de Pós-Graduação em Ciência da Computação, Belém, 2017.

LIMA, João Luís dos Reis. SOUZA, Rafael Ferreira de. **Um mapeamento sistemático da literatura em melhoria do processo de software e medição de software por propostas de gamificação e jogos sérios.** TCC - Universidade Federal de Santa Catarina. Florianópolis, 2018.

LIMA, Sheila Moutinho; VENDRAMEL, Wilson. **Mapeamento entre as práticas do Scrum e os processos do nível G do MPS.BR.** Fasci-Tech 2011. Disponível em: <<http://www.fatecsaocaetano.edu.br/fascitech/index.php/fascitech/articled/view/56/55>>. Acesso em: setembro de 2019.

MARTINS, Marcos Amâncio P. **Gestão Educacional: planejamento estratégico e marketing.** 1. ed. Rio de Janeiro: Brasport, 2007.

MATSUSHITA, Renan Shin Iti: 2010. **O impacto da integração entre o processo RUP com padrão PMBOK.** Faculdade de Tecnologia de São Caetano do Sul. Disponível em: <<http://www.fattocs.com.br/livro-apf/citacao/RenanSTMatsushita-2010.pdf>>. Acesso em: outubro de 2019.

MENAYO, J. M. T. and Ringach, E. R. (2006). **A guided approach to quality improvement of knowledge management practices**, Proceedings of I-KNOW, Vol. 6, pp. 6–8.

PMI, Project Management Institute. **Guia PMBOK - Um Guia do Conhecimento em Gerenciamento de Projetos**. 6ª edição. 2017.

PMI, Project Management Institute. **GUIA DE PRÁTICA ÁGIL**. 1ª edição. 2018.

_____, Project Management Institute. **O que é Gerenciamento de Projetos?** Disponível em: <<https://brasil.pmi.org/brazil/AboutUS/WhatIsProjectManagement.aspx>>. Acesso em: setembro de 2019.

PRESSMAN, R. S. **Engenharia de Software - uma abordagem profissional**. 7ª ed. v. I. Porto Alegre: AMGH Bookman, 2011.

ROCHA, Erico. **Como Usar a Análise SWOT Para Avaliar o Desempenho da Sua Equipe**. <<https://www.ignicaodigital.com.br/como-usar-analise-swot-para-avaliar-minha-equipe/>>. Acesso em: outubro de 2019.

RUBIN, Kenneth S. **Essential Scrum: A practical guide to the most popular Agile process**. Addison-Wesley, 2012.

RUP - **Rational Unified Process**. 2001. Disponível em: <https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf>. Acesso em: outubro de 2019.

SABBAGH, R. **Scrum: Gestão ágil para projetos de sucesso**. 1. ed. São Paulo: Casa do Código, 2013.

SCHEID, M., PESSOA, M. V., GOMES, R. F., RAIMUNDO, E. S., de OLIVEIRA, M.A. A., SANTOS, G., & ROCHA, A. R. **Implantação do MR-MPS Nível E no Centro de Computação da Aeronáutica de São José dos Campos**. São Paulo: ProQuality (UFLA), 2007.

SCHWABER, Ken, SUTHERLAND, Jeff. **The Scrum Guide**. Disponível em: <<https://www.scrumguides.org/scrum-guide.html>>. Acesso em: Setembro de 2019.

SCRUM PORTUGAL, 2019. **Scrum**. Disponível em: <<http://www.scrumportugal.pt/scrum/>>. Acesso em: outubro de 2019.

SEBRAE. **Ferramenta: 5W2H – Plano de Ação para Empreendedores.** Disponível em: <<http://www.sebrae.com.br/Sebrae/Portal%20Sebrae/Anexos/5W2H.pdf>>. Acesso em: outubro de 2019.

SEI. **Capability Maturity Model Integration (CMMI) for Development.** Version 1.3. USA: Carnegie Mellon, 2010.

SIQUEIRA, F. L. **Métodos Ágeis, 2004.** Disponível em: <http://www.levysiqueira.com.br/artigos/metodos_ageis.pdf>. Acesso em: setembro 2019.

SOFTEX. **Qualidade.** Disponível em: <<https://softex.br/mpsbr/>>. Acesso em: agosto de 2019.

_____. **Guia de Avaliação: 2017 - Parte 1** (abril de 2017). Disponível em: <<https://softex.br/mpsbr/guias/>>. Acesso em: agosto de 2019.

_____. **Guia de Implementação de Software – Parte 1: Nível G: 2016** (fevereiro de 2016). Disponível em: <<https://softex.br/mpsbr/guias/>>. Acesso em: agosto de 2019.

_____. **Guia Geral de Software: 2016** (janeiro de 2016). Disponível em: <<https://softex.br/mpsbr/guias/>>. Acesso em: agosto de 2019.

SOMMERVILLE, Ian. **Engenharia de Software.** 9a ed. São Paulo: Pearson Education Brasil, 2011.

SPINOLA, M. M., TONINI, A.C.; CARVALHO, M.M. Contribuição dos modelos de qualidade e maturidade na melhoria dos processos de software. EPUSP. **Revista Produção**, São Paulo, 2008.

VISUAL PARADIGM. **What is Planning Poker in Agile?.** Disponível em: <<https://www.visual-paradigm.com/scrum/what-is-agile-planning-poker/>>. Acesso em: outubro de 2019.

WYSOCKI, Robert K. **Effective project management: traditional, agile, extreme.** 7ª ed. John Wiley & Sons, 2014.

GLOSSÁRIO

Ágil - Um termo usado para descrever uma mentalidade de valores e princípios, conforme estabelecido no Manifesto Ágil.

Análise SWOT - identifica as oportunidades do projeto resultantes das forças da organização, assim como as ameaças decorrentes das fraquezas.

Daily Scrum – Também conhecido como *Daily Meetings*, é uma breve reunião de colaboração diária, na qual a equipe analisa o progresso do dia anterior, declara intenções para o dia atual e destaca os obstáculos encontrados ou previstos. (PMI, 2018).

Development Team – também conhecida como Equipe Dev, é uma equipe de desenvolvimento composta por profissionais que realizam o trabalho de fornecer um incremento potencialmente liberável do produto "Concluído" no final de cada *Sprint*. (SCHWABER e SUTHERLAND, 2017).

Framework - Um sistema básico ou estrutura de ideias ou fatos que apóiam uma abordagem. (PMI, 2018).

Incremento - Um produto funcional, testado e aceito que é um subconjunto do resultado geral do projeto. (PMI, 2018).

Iteração – Um ciclo de desenvolvimento com prazo de validade em um produto ou entrega, no qual todo o trabalho necessário para agregar valor é realizado. (PMI, 2018).

Kick-off - é uma reunião feita logo após a contratação do serviço, também utilizada para resolver os conflitos e obter o comprometimento tanto dos participantes internos do projeto quanto externos.

PMBOK – é um termo que remete a todo o conhecimento relacionado gerenciamento de projetos existente. Isso inclui práticas tradicionais comprovadas e amplamente aplicadas e práticas novas surgindo na profissão. (PMI, 2017)

Product Backlog - Uma lista ordenada de requisitos centrados no usuário que uma equipe mantém para um produto. (PMI, 2018).

Product Owner - Uma pessoa responsável por maximizar o valor do produto e que é, em última instância, responsável e responsável pelo produto final que é construído. (PMI, 2018).

RUP - é um compilado de melhores práticas para equipes de desenvolvimento de software, buscando refletir experiências e melhores práticas em evolução e comprovadas. (RATIONAL, 2001)

Scrum – Uma estrutura ágil para desenvolver e utilizar produtos complexos, com funções, eventos e artefatos específicos. (PMI, 2018).

Scrum Master - O coach da equipe de desenvolvimento e o proprietário do processo na estrutura Scrum. Remove obstáculos, facilita eventos produtivos e defende a equipe de interrupções. (PMI, 2018).

Scrum Team - é a Equipe *Scrum*, consiste em 3 funções, um *Product Owner*, um *Scrum Master* e um *Development Team*. (SCHWABER e SUTHERLAND, 2017).

Sprint - é o tempo de ciclo do desenvolvimento de um incremento, com durações variáveis, sendo um intervalo de um mês ou menos. (SCHWABER e SUTHERLAND, 2017).

Sprint Backlog - Uma lista de itens de trabalho identificados pela equipe do Scrum a serem concluídos durante o Sprint. (PMI, 2018).

Sprint Planning - Um evento colaborativo no Scrum no qual a equipe do Scrum planeja o trabalho para o sprint atual. (PMI, 2018).

Sprint Retrospective - Um workshop que ocorre regularmente, no qual os participantes exploram seu trabalho e resultados para melhorar o processo e o produto. (PMI, 2018).

Sprint Review – é uma Revisão da *Sprint* realizada no final da *Sprint* para inspecionar o Incremento e adaptar o *Backlog* do Produto. (SCHWABER e SUTHERLAND, 2017).

Stakeholders - grupo de interessados em algum projeto ou produto.

Template - um modelo a ser seguido, com uma estrutura predefinida.

Time boxes - Um período fixo de tempo, por exemplo, 1 semana, 1 quinzena, 3 semanas ou 1 mês. (PMI, 2018).

User Storys - Uma breve descrição do valor a entregar para um usuário específico. É uma promessa para uma conversa futura para esclarecer detalhes. (PMI, 2018).

APÊNDICE A - Formulário para entrevistas

Esse questionário tem o objetivo de gerar dados para avaliar o processo de desenvolvimento de software. Com essas respostas, posteriormente serão geradas propostas de melhoria de processo, do Gerenciamentos de projetos e Requisitos.

Pede-se que as questões sejam respondidas de forma completa e honesta, já que a identificação dos problemas internos é a primeira etapa para poder solucioná-los e melhorar o desempenho do setor, a qualidade do produto final e a qualidade de vida dos colaboradores.

Quanto à privacidade dos entrevistados, há um forte comprometimento em manter sigilo das respostas dadas. A intenção é coletar respostas sinceras, com críticas construtivas, sem comprometer nenhum colaborador.

As questões foram geradas baseando-se no Modelo de Referência MPS para Software. Para cada etapa e processamento do processo citada, deve-se destacar:

- Entradas, envolvidos, detalhamento e saídas (indicadores diretos).

Bom questionário!

1. O escopo do trabalho para o projeto é definido? Explique.
2. As tarefas e os produtos de trabalho definidos no escopo do projeto são decompostos em componentes menores e dimensionados? Explique.
3. O modelo de ciclo de vida do projeto é definido? Explique.
4. O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados? Explique.
5. O orçamento, o cronograma do projeto, marcos e pontos de controle são estabelecidos? São mantidos? Explique.
6. Os riscos do projeto são identificados? O seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados? São monitorados e atualizados? Explique.
7. Os recursos humanos e atividades do projeto são planejados considerando o perfil e o conhecimento necessários para

- executá-lo? Explique.
8. Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados? Explique.
 9. Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição? Explique.
 10. A organização tem um critério padrão para execução dessas atividades, isto deve está registrado no plano do projeto ou em outro documento? Explique.
 11. Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos? Explique.
 12. A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis? Explique.
 13. O Plano do Projeto é revisado com todos os interessados? Explique.
 14. O compromisso com o plano de projeto é obtido e mantido? Explique.
 15. O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado? Explique.
 16. Os recursos materiais e humanos são monitorados em relação ao planejado? Explique.
 17. Os dados relevantes do projeto são monitorados em relação ao planejado? Explique.
 18. Os riscos são monitorados em relação ao planejado? Explique.
 19. O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido? Explique.
 20. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento? Explique.
 21. Após o monitoramento, registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas? Explique.
 22. Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão? Explique.
 23. O entendimento dos requisitos é obtido junto aos fornecedores de requisitos? Explique.
 24. Os requisitos são avaliados com base em critérios objetivos?

Explique.

25. Um comprometimento da equipe técnica com estes requisitos é obtido? Explique.
26. A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida? Explique.
27. Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos? Explique.
28. As mudanças nos requisitos são gerenciadas ao longo do projeto? Explique.

APÊNDICE B - Tabela de avaliação dos Atributos de Processo

	API.1	AP 2.1				
	(i)	(i)	(ii)	(iii)	(iv)	(v)
GPR 1	SIM	SIM	SIM	SIM	SIM	SIM
GPR 2	SIM	SIM	SIM	SIM	SIM	SIM
GPR 3	SIM	SIM	SIM	SIM	SIM	SIM
GPR 4	NÃO	SIM	SIM	SIM	SIM	SIM
GPR 5	SIM	SIM	SIM	SIM	SIM	SIM
GPR 6	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO
GPR 7	SIM	SIM	SIM	SIM	SIM	SIM
GPR 8	SIM	SIM	SIM	SIM	SIM	SIM
GPR 9	SIM	SIM	SIM	SIM	SIM	SIM
GPR 10	SIM	SIM	SIM	SIM	SIM	SIM
GPR 11	SIM	SIM	SIM	SIM	SIM	SIM
GPR 12	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO
GPR 13	SIM	SIM	SIM	SIM	SIM	SIM
GPR 14	SIM	SIM	SIM	SIM	SIM	SIM
GPR 15	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO
GPR 16	SIM	SIM	SIM	SIM	SIM	SIM
GPR 17	SIM	SIM	SIM	SIM	SIM	SIM
GPR 18	SIM	SIM	SIM	SIM	SIM	SIM
GPR 19	SIM	SIM	SIM	SIM	SIM	SIM
GRE 1	SIM	SIM	SIM	SIM	SIM	SIM

GRE 2	NÃO	SIM	SIM	SIM	SIM	SIM
GRE 3	SIM	SIM	SIM	SIM	SIM	SIM
GRE 4	SIM	SIM	SIM	SIM	SIM	SIM
GRE 5	SIM	SIM	SIM	SIM	SIM	SIM
Porcentagem:	0,792	0,875	0,875	0,875	0,875	0,875
Avaliação do RAP:	L	T	T	T	T	T
Avaliação do AP:	L	T				

ANEXO A - Resumo do Guia de Avaliação

Esse apêndice busca destacar e detalhar as partes relevantes para o trabalho do Guia de Avaliação - Parte I – Processo e Método de Avaliação MA-MPS.

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

- permitir a avaliação objetiva dos processos de software, de serviços ou de gestão de pessoas de uma organização/unidade organizacional;
- permitir a atribuição de um nível de maturidade do MR-MPS-SW 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 Processo e Método de Avaliação MA-MPS é verificar a maturidade da unidade organizacional na execução de seus processos de software, serviços ou gestão de pessoas. Uma avaliação pode ser realizada em uma organização inteira ou em uma unidade organizacional, que faz parte desta organização. No caso, a unidade organizacional é o projeto.

Para que uma avaliação seja conduzida com sucesso, é necessário: comprometimento do patrocinador; motivação da liderança e equipe; fornecimento de feedback; confidencialidade; percepção dos benefícios; credibilidade.

Como resultado da execução deste processo:

- São obtidos dados e informações que caracterizam os processos de software unidade organizacional;
- É determinado o grau em que os resultados esperados e atributos de processos são alcançados e os processos atingem o seu propósito;
- É atribuído o nível de maturidade à organização/unidade organizacional.

CONSIDERAÇÕES RELACIONADAS À ESCOPO DA

AVALIAÇÃO

Durante o planejamento da avaliação, o primeiro passo é definir o escopo da avaliação que consiste em definir:

- i. A unidade organizacional, isto é, a(s) parte(s) da organização que será(ão) avaliada(s).
- ii. Os modelos MPS que serão avaliados na unidade organizacional.
- iii. Os processos que serão avaliados em cada modelo.
- iv. O nível MPS de cada modelo que será considerado na avaliação.

CONSIDERAÇÕES RELACIONADAS À SELEÇÃO DAS PESSOAS A SEREM ENTREVISTADAS

A seleção das pessoas a serem entrevistadas é realizada ao se elaborar o Plano de Avaliação e deve estar concluída antes da avaliação final. Esta seleção está diretamente relacionada aos processos e projetos/serviços/áreas que serão avaliados. É possível selecionar pessoas de outros projetos/serviços/áreas para se aprofundar em uma parte específica da avaliação, entretanto os projetos/serviços/áreas que serão avaliados devem estar representados em todas as entrevistas, sempre que possível.

As entrevistas podem ser agendadas para serem conduzidas individualmente ou em grupos. Os grupos são representados por diferentes pessoas que executam o mesmo tipo de tarefa (papéis) no mesmo projeto/serviço/área ou em diferentes projetos/serviços/áreas. O melhor é conseguir envolver o máximo possível de pessoas dos projetos/serviços/áreas nas entrevistas. Deve-se cuidar para que, em um grupo, não se tenha nenhum membro que seja superior hierárquico de outros membros do grupo.

CONSIDERAÇÕES RELACIONADAS À COLETA DE INDICADORES

Os indicadores de implementação podem ser de três (3) tipos:

1. Indicadores diretos: são o objetivo de uma tarefa, ou seja, o produto principal da realização de uma tarefa.
2. Indicadores Indiretos: são artefatos que são consequência da realização de uma tarefa e que referendam a implementação de um resultado, mas que não são o objetivo da realização da tarefa e normalmente são gerados durante a execução dos projetos/serviços ou nas áreas no caso de gestão de pessoas.
3. Afirmações: são obtidas em entrevistas e/ou apresentações e

confirmam a implementação do processo, seus resultados e atributos.

Apenas os indicadores diretos e as afirmações são obrigatórios. Inicialmente, somente os indicadores diretos devem obrigatoriamente fazer parte da planilha, pois as afirmações serão coletadas durante as entrevistas e/ou apresentações na avaliação final.

A inclusão de indicadores indiretos é opcional e podem ser incluídos caso a unidade organizacional considere que podem auxiliar no entendimento da implementação. Em alguns casos pode ser desejável incluir na planilha a descrição dos processos, que entretanto não são considerados indicadores indiretos.

CONSIDERAÇÕES RELACIONADAS ÀS ENTREVISTAS

As entrevistas são um dos mais importantes componentes de uma avaliação, para ajudar a equipe de avaliação a entender e analisar como a unidade organizacional implementar os seus processos. As entrevistas também ajudam a identificar o que pode ser melhorado e o nível de institucionalização dos processos.

As entrevistas devem ser realizadas em uma sala fechada, com privacidade suficiente para que todos possam se expressar com liberdade. Para que uma entrevista tenha o resultado esperado é necessário que:

- Todos sejam tratados com respeito;
- Exista um clima onde as pessoas se expressem livremente e todos participem;
- Não existam interrupções, julgamentos e nem observações ao que é dito (por exemplo, não se deve expressar julgamento: “OK”, “certo”, “errado”); e
- Sejam evitados tensões e constrangimentos.

Durante uma entrevista, toda a equipe de avaliação, principalmente a miniequipe encarregada dos processos que são o foco da entrevista, deve tomar notas do que está sendo dito e que é importante para a avaliação. Todas as notas devem registrar data, hora e o objetivo da entrevista, e devem ser destruídas ao final da avaliação para preservar a confidencialidade. O avaliador líder poderá não tomar notas, deixando essa tarefa para o restante da equipe, se achar conveniente.

As questões (perguntas) para as entrevistas devem ser elaboradas durante a verificação das evidências, mas deverão ser revistas antes de cada entrevista, para que a necessidade de cobertura da avaliação seja atendida e para que perguntas não sejam repetidas sem necessidade. Sugere-se que as entrevistas devem ter a duração de no máximo 1 hora (60 minutos), a fim de não se tornarem cansativas e improdutivas.

Primeiramente explica o objetivo da avaliação e da entrevista.

Pede aos entrevistados que se apresentem, dizendo o nome, o projeto do qual participaram, a função no projeto e o tempo de trabalho na unidade organizacional.

Questões sobre a implementação dos processos e atributos de processo e perguntas finais sobre a unidade organizacional.

A cobertura das afirmações está relacionada com a necessidade de se confirmar que o resultado foi alcançado nos projetos. Para isso, deve haver a seguinte cobertura de afirmações para cada processo:

- Para cada resultado esperado deve haver pelo menos uma afirmação;

CONSIDERAÇÕES RELACIONADAS AO REGISTRO DAS AFIRMAÇÕES

Após cada entrevista a equipe de avaliação deve consolidar os dados das entrevistas. Como a quantidade de informações produzidas pelas entrevistas é grande, a equipe de avaliação deve resumir na planilha, na forma de declarações precisas, as suas observações sobre o que realmente foi afirmado, os pontos fortes, os pontos fracos e as oportunidades de melhoria. Para manter a confidencialidade das entrevistas, a fonte de informação deve ser apagada da planilha ao final da avaliação.

O objetivo da consolidação é, também, controlar o andamento da avaliação, permitindo verificar se existem dados suficientes para a equipe fazer um julgamento, ou se são necessárias mais entrevistas para, com mais informações, ser possível o julgamento.

CONSIDERAÇÕES RELACIONADAS À CARACTERIZAÇÃO DE ATRIBUTO DE PROCESSO

A caracterização dada ao AP 1.1 é uma apreciação consolidada dos resultados esperados do processo. O grau T em AP 1.1 significa um alcance de mais do que 85% do conjunto dos resultados esperados do processo, sem atribuição de “Parcialmente Implementado” (P) ou “Não Implementado” (N) a resultados do processo. Analogamente, os graus L, P e N refletem o alcance do conjunto dos resultados esperados do processo, segundo os percentuais da Tabela 9.

A porcentagem de orientação para caracterização de um atributo de processo deve ser vista de forma global no que se refere ao grau de implementação do atributo de processo e é, ao mesmo tempo, uma avaliação qualitativa e quantitativa.

CONSIDERAÇÕES RELACIONADAS À DOCUMENTAÇÃO DE PONTOS FORTES, PONTOS FRACOS E OPORTUNIDADES DE MELHORIA.

Nesse momento, também, são documentadas as observações que são classificadas em: pontos fortes, pontos fracos e oportunidades de melhoria. Essas observações representam o entendimento da equipe de avaliação sobre o que foi avaliado. Uma observação é uma assertiva sobre a implementação de um resultado, acordada pela equipe como um todo, utilizando a técnica do consenso. Uma observação pode estar relacionada ou não ao MR-MPS-SW, mas sempre identifica alguma questão que tem um impacto na maturidade do processo de software da unidade organizacional. Toda caracterização diferente de T deve ter associado um ponto fraco. Oportunidades de melhoria podem estar associadas a caracterizações de resultados de processo ou a atributos de processo como T, L, P ou N.

ANEXO B - Resumo do Guia de Implantação

Esse apêndice busca destacar e detalhar as partes relevantes para o trabalho do Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS-SW:2016.

1. A implantação do nível G

As atividades e tarefas necessárias para atender ao propósito e aos resultados esperados não são definidas no modelo, devendo ficar a cargo dos usuários do MR-MPS-SW.

O Guia de Implementação - Parte 1 fornece orientações e fundamentação para Implementação do Nível G do MR-MPS-SW nas organizações, detalhando os processos contemplados no respectivo nível de maturidade e os resultados esperados com a implementação dos processos.

No nível G, o projeto pode usar os seus próprios padrões e procedimentos, não sendo necessário que se tenha padrões organizacionais. Se, porventura, a organização possuir processos já definidos e os projetos necessitam se adaptar os processos existentes, deve-se registrar essa adaptação durante o planejamento do projeto. Adaptações podem incluir alteração em processos, atividades, ferramentas, técnicas, procedimentos, padrões, medidas, dentre outras.

Diversas organizações de software trabalham com evolução de produtos e precisam adequar as suas formas de trabalhar para se tornarem organizações orientadas a projetos. Ser orientada a projetos significa redefinir algumas operações (atividades de rotina) já em andamento, como projeto, estabelecendo objetivos, prazos e escopo para sua execução.

A seguir, são melhor definidos os Resultados Esperado de Processo e os Resultados de Atributos de Processos:

- GPR 1. O escopo do trabalho para o projeto é definido;
Verificar se resposta abrange: definição do objetivo e da motivação, os limites e restrições, todos os produtos que serão entregues e os outros produtos gerados pelo projeto, entre outras informações.
- GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;
Os principais produtos de trabalho são decompostos em componentes menores em uma estrutura estabelecida e apropriada de

decomposição do trabalho. Visualização do tamanho dos produtos (dimensão das funcionalidades sob o ponto de vista do usuário).

São contadas tabelas internas e externas ao sistema, classes, objetos, relatórios, telas, consultas a banco de dados, cálculos, transações e atores dos casos de uso, linhas de código etc. No nível G, a estimativa de escopo, produtos e tarefas pode ser feita baseada na complexidade, no número de requisitos ou no uso da EAP juntamente com dados históricos e a experiência em projetos anteriores.

- GPR 3. O modelo e as fases do ciclo de vida do projeto são definidos;

Está determinado um modelo de ciclo de vida que melhor se adapta ao projeto como esqueleto do processo e suas fases. Os principais modelos de ciclo de vida podem ser agrupados em categorias: modelos sequenciais ou cascata, incrementais, iterativos, adaptativos e extremos.

- GPR 4. (Até 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;

As estimativas de esforço e custo são, normalmente, baseadas nos resultados de análises utilizando modelos e/ou dados históricos aplicados ao tamanho, atividades e outros parâmetros de planejamento.

As estimativas de esforço e custo tipicamente consideram: o escopo, produtos de trabalho e as tarefas estimadas para o projeto; os riscos; as mudanças já previstas; o ciclo de vida escolhido para o projeto; viagens previstas; nível de competência da equipe do projeto, dentre outros.

- GPR 5. O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos;

As dependências entre tarefas são estabelecidas e potenciais gargalos são identificados utilizando métodos apropriados. Recursos requeridos são refletidos nos custos estimados. Cronograma é estabelecido. Marcos e pontos de controle são definidos. É importante ter-se o cuidado de manter a coerência entre ciclo de vida, EAP, estimativas e cronogramas.

- GPR 6. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;

Projetos têm riscos e estes precisam ser identificados, analisados e priorizados. A análise da probabilidade de ocorrência e da gravidade dos problemas decorrentes de sua ocorrência ajuda a definir a prioridade dos riscos. Uma planilha de riscos, contendo dados como identificador, descrição, probabilidade, impacto e prioridades no seu tratamento, pode ser utilizada para identificação dos riscos, monitoração dos riscos identificados e atualização da lista de riscos. É importante demonstrar que esta planilha está sendo monitorada e atualizada.

- GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;

O planejamento de recursos humanos inclui informações de como e quando o recurso será envolvido no projeto, critérios para sua liberação, competência necessária para a execução das atividades, mapa de competências da equipe e identificação de necessidades de treinamento. São também mantidos registros das necessidades e disponibilidade de recursos humanos.

Este resultado esperado possui dois pontos principais:

- (i) planejamento prévio das necessidades de pessoal em relação a competências (conhecimento, habilidades, atitudes e experiências);
- (ii) a alocação dos recursos humanos ao projeto de acordo com o planejamento realizado.

- GPR 8. (Até o nível F) Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;

Este resultado faz referência à necessidade de se planejar, com base na EAP (ou estrutura equivalente), as tarefas e previstos os recursos e o ambiente necessários, incluindo, por exemplo, equipamentos, ferramentas, serviços, componentes, viagens e requisitos de processo (processos especiais para o projeto). Este resultado é importante porque recursos especiais precisam de orçamento e planejamento de sua aquisição, o que pode se tornar crítico em alguns projetos.

- GPR 9. Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;

Os dados do projeto são as várias formas de documentação exigidas para sua execução. Os dados podem estar em qualquer formato e existir em qualquer meio, A distribuição pode ocorrer de várias formas,

incluindo a transmissão eletrônica. A identificação, coleta, armazenamento, distribuição (incluindo regras de segurança e confidencialidade) para garantir a integridade, acesso e segurança aos dados devem ser planejados.

Se a organização tem um critério padrão para execução dessas atividades, isto deve ser registrado no plano do projeto ou em outro documento.

- GPR 10. Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos;

A realização do planejamento do projeto é garantida pelos resultados esperados no escopo do nível G do MR-MPS-SW do processo Gerência de Projetos (GPR), que prevê, dentre outros, a criação do cronograma de atividades, o planejamento de recursos humanos, custos, riscos, dados etc. A reunião destes documentos é entendida como sendo o Plano de Projeto.

- GPR 11. A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados;

O estudo de viabilidade considera o escopo do projeto e examina aspectos técnicos (requisitos e recursos), financeiros (capacidade da organização) e humanos (disponibilidade de pessoas com a capacitação necessária). Pode-se considerar também os objetivos de negócio e a composição do portfólio de projetos da organização.

- GPR 12. O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido;

Deve-se revisar o planejamento com eles e conciliar as diferenças existentes entre os recursos estimados e disponíveis. É importante, para minimizar os riscos dos projetos, que o comprometimento dos envolvidos seja obtido antes da participação na execução de alguma tarefa.

- GPR 13. O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado;

A aderência aos diversos planos deve ser avaliada continuamente durante todo o ciclo de vida do projeto. O objetivo deste resultado esperado é assegurar que haja monitoração do projeto em relação a aspectos relacionados às tarefas, estimativas, orçamento e cronograma

(ver GPR2, GPR3, GPR4, GPR5). Em geral, durante um monitoramento, faz-se uma análise do que foi planejado anteriormente com os valores atuais das variáveis consideradas.

- GPR 14. Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado; O objetivo deste resultado esperado é garantir que o projeto seja monitorado em relação aos itens planejados referentes a recursos materiais e recursos humanos (ver GPR7 e GPR8).

- GPR 15. Os riscos são monitorados em relação ao planejado No decorrer do projeto novos riscos podem ser identificados para o projeto e os parâmetros dos riscos já identificados podem ser alterados (ver GPR6).

- GPR 16. O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido; Devem ser identificados os interessados relevantes no projeto, em que fases eles são importantes e como eles serão envolvidos (comunicações, revisões em marcos de projeto, comprometerimentos, entre outros) (ver GPR7). Uma vez identificado e planejado o envolvimento, este deverá ser seguido, monitorado e mantido ao longo de todo o projeto.

- GPR 17. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento; Revisões em marcos do projeto não devem ser confundidas com o acompanhamento descrito em GPR13, GPR14 e GPR15, que é derivado do acompanhamento do dia-a-dia do projeto. Os marcos do projeto precisam, portanto, ser previamente definidos ao se realizar o planejamento do projeto. Este resultado é importante porque as revisões em marcos são oportunidades para verificar, de forma ampla, o andamento de todo o projeto, independente do acompanhamento do dia-a-dia.

- GPR 18. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas; As atividades de revisão em marcos (GPR17) e de monitoramento (GPR13, GPR14 e GPR15) do projeto possibilitam a identificação de problemas que estejam ocorrendo nos projetos. É natural que problemas e desvios em relação ao planejamento aconteçam durante

a execução dos projetos. Estes problemas e desvios devem ser analisados e registrados, por exemplo, por meio de ferramentas específicas, planilhas ou outros tipos de mecanismos de gerenciamento de problemas.

- GPR 19. Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão;

Como resultado do acompanhamento do projeto (GPR13, GPR14 e GPR15) e das revisões em marcos (GPR17), problemas são identificados, analisados e registrados (GPR18). Para completar o trabalho de monitoramento do projeto, os problemas precisam ser corrigidos e gerenciados até a sua resolução, com base em planos de ações, estabelecidos especificamente para resolver os problemas levantados e registrados (GPR19).

- GRE 1. O entendimento dos requisitos é obtido junto aos fornecedores de requisitos;

O objetivo deste resultado é garantir que os requisitos estejam claramente definidos a partir do entendimento dos requisitos realizado junto aos fornecedores de requisitos. Informações sobre esses fornecedores podem ser identificadas no plano do projeto, bem como informações sobre como será a comunicação com eles. Essas comunicações devem ser registradas formalmente em atas, e-mails, ferramentas de comunicação ou outros meios.

- GRE 2. Os requisitos são avaliados com base em critérios objetivos e um comprometimento da equipe técnica com estes requisitos é obtido;

Em geral, é aconselhável que os requisitos sejam avaliados pela equipe técnica antes de serem submetidos para aprovação pelo cliente para evitar retrabalho ou a apresentação de um documento sem qualidade técnica adequada para o cliente. Além disso, um comprometimento formal da equipe técnica com os requisitos deve ser obtido e registrado, por exemplo, na forma de ata de reunião, e-mail ou algum outro mecanismo.

- GRE 3. A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;

Este resultado indica a necessidade de se estabelecer um mecanismo que permita rastrear a dependência entre os requisitos e os

produtos de trabalho. Ter definida a rastreabilidade facilita a avaliação do impacto das mudanças de requisitos que possam ocorrer, por exemplo, nas estimativas do escopo, nos produtos de trabalho ou nas tarefas do projeto descritas no cronograma.

- GRE 4. Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;

Este resultado sugere, portanto, a realização de revisões ou de algum mecanismo equivalente para identificar inconsistências entre os requisitos e os demais elementos do projeto como, por exemplo, planos, atividades e produtos de trabalho.

- GRE 5. Mudanças nos requisitos são gerenciadas ao longo do projeto.

As necessidades de mudanças devem ser registradas e um histórico das decisões acerca dos requisitos deve estar disponível. Estas decisões são tomadas por meio da realização de análises de impacto da mudança no projeto e podem incluir aspectos como: influência em outros requisitos, expectativa dos interessados, esforço, cronograma, riscos e custo. É importante destacar que o mecanismo de rastreabilidade bidirecional (GRE 3) instituído é um importante mecanismo para facilitar a análise de impacto.

- AP1.1 - O processo é executado

(i) O processo produz os resultados definidos;

Avalia a eficácia dos processos. Este atributo evidencia o quanto o processo atinge o seu propósito.

- AP 2.1 A execução do processo é gerenciada

(i) existe uma política organizacional estabelecida e mantida para o processo;

Avalia se existe políticas de execução das tarefas, padrões de execução.

(ii) a execução do processo é planejada;

O planejamento deve incluir identificação e disponibilização dos recursos e informações necessárias para a execução do processo, definição, atribuição e comunicação das responsabilidades pela execução do processo e planejamento da comunicação entre as partes interessadas. Avalia se o planejamento do processo de execução das tarefas.

(iii) a execução do processo é monitorada em relação ao planejado e, quando necessário, ajustes são realizados;

Avalia Monitoramento da execução do processo

(iv) as pessoas que executam o processo estão preparadas para executar suas responsabilidades;

Assegurar que as pessoas alocadas tenham as habilidades, conhecimentos e experiências necessários para executar ou apoiar o processo.

(v) as atividades, o status e os resultados do processo são revistos com a gerência de nível superior e são tratadas questões críticas;

Fornecer visibilidade à alta gerência com relação ao estado da execução dos processos, considerando sua adequação, operação com recursos apropriados e alcance dos resultados esperados.

ANEXO C - Resumo do Guia de utilização do GitLab

1. Introdução

Este documento tem como objetivo resumir o Guia da ferramenta utilizada no projeto. Nele é realizado a padronização de preenchimento das tarefas a fim de que se possam verificar as atividades em execução e aprimorar os processos em andamento.

2. O que é o GitLab

O GitLab é um gerenciador de repositório de software GIT que integra o gerenciamento e controle das tarefas dos projetos. Ele também provê funcionalidades que permitem aos usuários vincularem os commits realizados com as tarefas (issues).

A ferramenta contempla todas as fases do ciclo de desenvolvimento de produtos de software, podendo ser adaptada/configurada para a utilização em conjunto com o framework Scrum.

As issues são as unidades de trabalho dentro do GitLab. Elas podem representar os épicos, os temas, as histórias de usuário e as atividades que serão cadastradas. Nelas, serão vinculadas labels e milestones, que serão melhor explicadas nas próximas sessões. Além disso, nelas, são registrados os tempos estimados e efetivamente gastos para a execução de determinada tarefa.

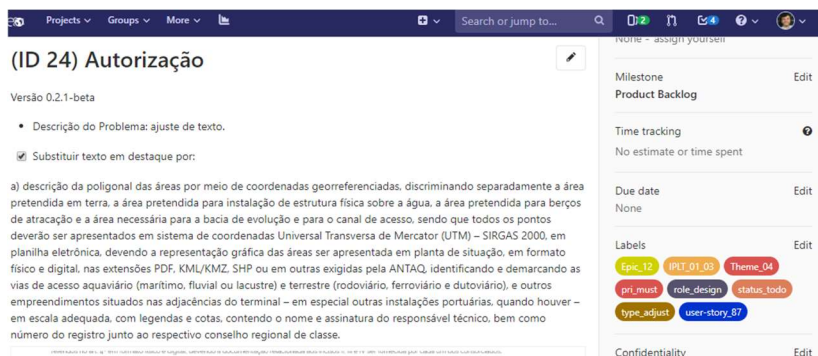


Figura 1 - Exemplo de issue (tarefa) no Gitlab

3. Organização do projeto

Os projetos deverão ser organizados no GitLab de acordo com a seguinte estrutura hierárquica:

Parceiro > Projeto > Subprojeto 1
 Parceiro > Projeto > Subprojeto 2

Devem ser definidos de acordo com a arquitetura de cada solução. A Figura 2 fornece um exemplo da organização de um projeto web que se utiliza de uma API para a camada de integração com o sistema gerenciador de banco de dados.

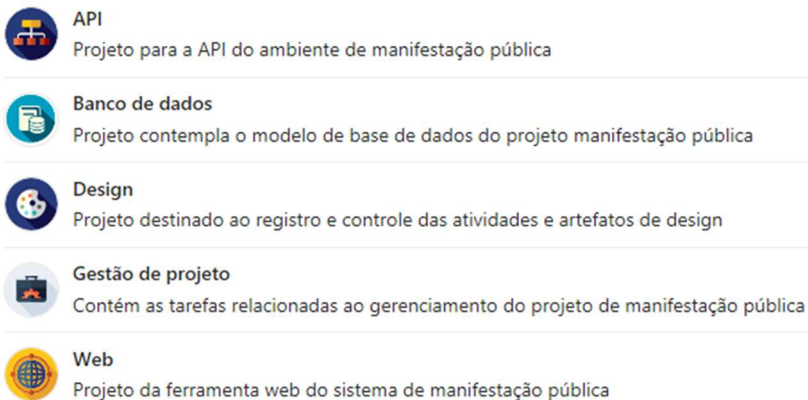


Figura 2 - Exemplo de organização dos projetos

4. Gestão de produto

No Gitlab, a gestão de produto destina-se ao cadastro e controle dos temas, épicos e histórias de usuários e técnicas que compõem o escopo da solução tecnológica a ser desenvolvida, bem como os artefatos gerados durante o período de análise e levantamento de requisitos e especificação das funcionalidades, além de atividades relacionadas ao gerenciamento do projeto.

Theme: é utilizado para agrupar épicos e histórias com base em algum objetivo de negócio ou uma definição de itens do roadmap. Pode ser utilizado para refletir o escopo macro do produto com as grandes entregas em um cronograma, por exemplo.

Epic: são histórias que representam itens grandes demais ou sem detalhes suficientes para serem desenvolvidos. Somente necessitarão de mais detalhes quando ganharem prioridade.

User story: representa uma funcionalidade ou característica do produto “narrada” pelo ponto de vista do usuário (papel que interage com o produto). Deve ser uma forma para se atingir a visão do produto.

As histórias de usuário devem ser elaboradas segundo a regra de formação a seguir:

[HISTÓRIA <número sequencial três dígitos>] Como um <papel>, posso/gostaria/devo <função> para/de <resultado para o negócio>

Exemplo:

[HISTÓRIA 001] Como um analista, gostaria de realizar a validação de regras topológicas para identificar possíveis problemas de sobreposição

Toda história de usuário deve ser associada à documentação técnica utilizando a Unified Modeling Language (UML). A especificação técnica da funcionalidade é feita por meio da ferramenta Enterprise Architect (EA) e exportada em PDF utilizando o *template* padrão estabelecido. O documento PDF exportado deve ser enviado (commit e push) para o projeto no GIT e deve-se referenciar no commit a história à qual o documento está relacionado. Desta maneira, o desenvolvedor tem fácil acesso à documentação da história de usuário.

5. Labels

Toda issue deve conter uma ou mais labels associadas. Elas são utilizadas para identificar características das tarefas como status, tipo, prioridade entre outras. O GitLab permite que a listagem das issues seja filtrada por estas labels. A funcionalidade de board também utiliza as labels para apresentar cada uma das listagens.

O processo de desenvolvimento utiliza-se das labels para identificar cada etapa do ciclo de vida da tarefa. Com base nas necessidades de negócio levantadas pela equipe, foram identificadas e padronizadas uma série de labels que deverão ser utilizadas em todos os projetos desenvolvidos.

a. Prioridade

A prioridade das issues deverá ser classificada de acordo com o método MoSCoW e as labels disponíveis podem ser visualizadas na Tabela 1.

Tabela 1 - Tabela das labels de prioridade

Label	Descrição
pri_must	Descreve um requisito que deve ser atendido na solução final para que tal solução seja considerada um sucesso.
pri_should	Trata-se frequentemente de um requisito crítico que pode ser atendido de outras formas se for estritamente necessário.
pri_could	Descreve um requisito que é considerado desejável, mas não necessário, e que será incluído caso o tempo e os recursos permitam.
pri_wont	Representa um requisito que as partes interessadas concordam em não implementar. A tarefa deve ser fechada.

b. Papel

Os papéis representam a etapa corrente e os responsáveis pela execução da atividade. Essa informação é utilizada para identificar qual área é responsável pela execução de determinada etapa do ciclo de vida da tarefa.

Tabela 2 - Tabela das labels dos papéis

Label	Descrição
role_business	Análise de negócio
role_analysis	Análise de sistema
role_design	Design
role_dev	Desenvolvimento

role_qa	Garantia da qualidade
role_ops	Operação
role_db	Banco de dados
role_review	Review

c. Status

Os status informam a situação atual a qual a tarefa se encontra. Eles serão alterados conforme a issue progride no seu desenvolvimento.

Tabela 3 - Tabela das labels dos status

Label	Descrição
status_todo	Tarefa que está aguardando para ser iniciada.
status_blocked	Tarefa bloqueada por motivos externos.
status_done	Tarefa que já foi finalizada e, se for o caso, testada.
status_discarded	Tarefa que foi avaliada e que não será executada.
status_doing	Tarefa que está sendo executada.

d. Tipos

A label do tipo identifica se uma tarefa é referente a um problema no sistema (bug) ou se é uma nova funcionalidade. A Tabela 4 apresenta os tipos de issues mapeados e o padrão de nomenclatura de seus labels.

Tabela 4 - Tabela das labels dos tipos

<i>Label</i>	Descrição
--------------	-----------

<i>type_bug</i>	Uma condição acidental que faz com que o sistema não consiga executar sua função.
<i>type_new</i>	Novas funcionalidades.
<i>type_adjust</i>	Correções ou melhorias que não impactam no funcionamento do sistema.
<i>type_check</i>	Atividade que necessita ser averiguada.

Uma vez definidas e priorizadas as histórias (*product backlog*), deve-se agrupá-las em *milestones* para que seu desenvolvimento seja planejado.

e. Revisão das tarefas fechadas

As *milestones* podem ser utilizadas para definir uma versão, *Sprint* ou qualquer disponibilização. O GitLab permite que sejam criadas *milestones* nos projetos ou nos grupos. Criando-se as *milestones* com o mesmo título nos projetos e no grupo, a ferramenta apresenta, no grupo, uma visão consolidada da *milestone*, fornecendo o status do andamento das atividades, tal qual a Figura 6.

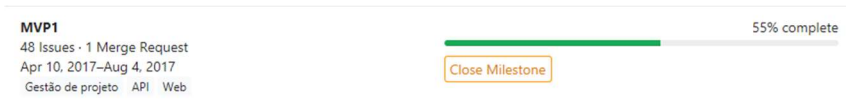


Figura 6 - Exemplo de milestone

A definição da milestone deve ser feita pelo product owner. Após esta definição, deverá ocorrer a cerimônia de *Sprint* planning com a equipe do projeto para que as tarefas da *Sprint* sejam planejadas. Nela, o product owner deve dar uma visão geral de cada uma das histórias definidas para a milestone e os membros do time deverão elencar as atividades necessárias para o desenvolvimento das histórias. Além disso, deve-se utilizar esse momento para estimar as atividades e cadastrar essas informações na ferramenta.

ANEXO D - *Template* do Plano de Trabalho

1. DADOS CADASTRAIS

Empresa				CNPJ	
Endereço					
Cidade	UF	CEP	DDD/Telefone	EA	
Conta Corrente		Banco	Agência	Praça de Pagamento	
Nome do Responsável				CPF	
Endereço				CEP	

2. CLIENTE

Órgão/Entidade		CNPJ	
Nome do Responsável		CPF	
Endereço		CEP	

3. DESCRIÇÃO DO PROJETO

3.1	Título do Projeto	Período de Execução	
		Início	Término
3.2	Justificativa da Proposição		
3.3	Objetivo Geral do Projeto		

4. ATIVIDADES

4.1 Estrutura analítica do projeto

I. Atividade 1

<Descrição da atividade 1>

II. Atividade 2

<Descrição da atividade 2>

5. CRONOGRAMA DOS OBJETOS

Nessa seção é apresentado um Diagrama de Gantt com cronograma de atividades e marcos.

6. PLANO DE APLICAÇÃO

6.1 Orçamento Detalhado

Discriminação	Qtde.	Período	Custo/mês	Custo total
Analista de negócio			R\$	R\$
Analista de sistemas			R\$	R\$
Programador Jr.			R\$	R\$
Designer			R\$	R\$
Tester			R\$	R\$
Material de consumo			R\$	R\$
Material gráfico			R\$	R\$
Diárias			R\$	R\$
Passagens aéreas			R\$	R\$

Telefone internet			R\$	R\$
Infraestrutura			R\$	R\$
Subtotal				R\$
Despesas operacionais diversas				
				R\$
				R\$
TOTAL DO PROJETO				

6.2 Cronograma de Desembolso

DESCRIÇÃO	PRAZO DE CUMP. (EM MESES)	FÍSICO		FINANCEIRO		
		UNID.	QUANT	N. DA PARC.	LIBER. (MÊS)	VALOR (EM R\$ 1,00)
TOTAL						

7. DECLARAÇÃO

Declaração...

Local, data .

Data

Empresa

8. APROVAÇÃO PELO CLIENTE

Local e Data

Cliente

ANEXO E - *Template* de Sprint de projeto

SEMANA 1							
SPRINT (N)	PERIODO	PAPEL	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5
Manhã	SCRUM MASTER	Monitoramento e apoio a equipe	Monitoramento e apoio a equipe Apolo ao P.O.	Monitoramento e apoio a equipe Apolo ao P.O.	Monitoramento e apoio a equipe Apolo ao P.O.	Monitoramento e apoio a equipe Apolo ao P.O.	Monitoramento e apoio a equipe Apolo ao P.O.
	PRODUCT OWNER	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog
	DEV TEAM	Desenvolver v.0.0.n-rc1	Desenvolver v.0.0.n-rc1	Desenvolver v.0.0.n-rc1 PUBLICAR PROD - v.0.0.(n-1)	Desenvolver v.0.0.n-rc1	Desenvolver v.0.0.n-rc1	Desenvolver v.0.0.n-rc2
	TESTERS	Testar versão v.0.0.(n-1)	Testar versão v.0.0.(n-1)	Testar versão v.0.0.(n-1)	Testar versão v.0.0.(n-1)	Testar versão v.0.0.(n-1)	Testar versão v.0.0.n-rc1
	GERAL						
Tarde	SCRUM MASTER	Monitoramento e apoio a equipe	Monitoramento e apoio a equipe Apolo ao P.O.	Monitoramento e apoio a equipe Apolo ao P.O.	Monitoramento e apoio a equipe Apolo ao P.O.	Monitoramento e apoio a equipe Apolo ao P.O.	Monitoramento e apoio a equipe Apolo ao P.O.
	PRODUCT OWNER	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	REUNIÃO - ALINHAMENTO COM STAKEHOLDERS Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog
	DEV TEAM	Desenvolver v.0.0.n-rc1	Desenvolver v.0.0.n-rc1	Desenvolver v.0.0.n-rc1	Desenvolver v.0.0.n-rc1	PUBLICAR DEV - v.0.0.n-rc1	Desenvolver v.0.0.n-rc2
	TESTERS	Testar versão v.0.0.(n-1)	Testar versão v.0.0.(n-1)	Testar versão v.0.0.(n-1)	Testar versão v.0.0.(n-1)	Testar versão v.0.0.(n-1)	Testar versão v.0.0.n-rc1
	GERAL						

		SEMANA 2				
		Dia 6	Dia 7	Dia 8	Dia 9	Dia 10
PERÍODO	PAPEL	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
Manhã	SCRUM MASTER	Monitoramento e apoio a equipe Apoio ao P.O.	Monitoramento e apoio a equipe Apoio ao P.O.	Monitoramento e apoio a equipe Apoio ao P.O.	Monitoramento e apoio a equipe Apoio ao P.O.	Monitoramento e apoio a equipe Apoio ao P.O.
	PRODUCT OWNER	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Priorização de bugs para fase de correções Priorização do Próximo Sprint (Sprint Backlog)	Priorização de bugs para fase de correções Priorização do Próximo Sprint (Sprint Backlog)	Priorização do Próximo Sprint (Sprint Backlog)
	DEV/TEAM	Desenvolver v.0.0.n-rC2	Desenvolver v.0.0.n-rC2	Correções de bugs encontrados na v.0.0.n-rC1	Correções de bugs encontrados v.0.0.n-rC1 e rC2	Correções de bugs encontrados v.0.0.n-rC1 e rC2 PUBLICAR DEV/HOMI - v0.0.n-rC3/v.0.0.n
	TESTERS	Testar versão v.0.0.n-rC1	Testar versão v.0.0.n-rC1	Testar versão v.0.0.n-rC2	Testar versão v.0.0.n-rC2	
	GERAL					Revisão do Sprint Backlog
Tarde	SCRUM MASTER	Monitoramento e apoio a equipe Apoio ao P.O.	Monitoramento e apoio a equipe Apoio ao P.O.	Monitoramento e apoio a equipe Apoio ao P.O.	Monitoramento e apoio a equipe Apoio ao P.O.	SPRINT PLANNING Feedback dos Stakeholders Revisão do Sprint executado Apresentação do próximo Sprint Dimensionamento do Sprint (estimate) Aceitação do Sprint
	PRODUCT OWNER	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Especificação de novas funcionalidades Priorização de bugs para fase de correções Refinamento de Backlog	Priorização de bugs para fase de correções Priorização do Próximo Sprint (Sprint Backlog)	COM STAKEHOLDERS Priorização de bugs para fase de correções Priorização do Próximo Sprint (Sprint Backlog)	
	DEV/TEAM	Desenvolver v.0.0.n-rC2	Desenvolver v.0.0.n-rC2 PUBLICAR DEV - v0.0.n-rC2	Correções de bugs encontrados v.0.0.n-rC1 e rC2	Correções de bugs encontrados v.0.0.n-rC1 e rC2	
	TESTERS	Testar versão v.0.0.n-rC1	Testar versão v.0.0.n-rC1	Testar versão v.0.0.n-rC2	Testar versão v.0.0.n-rC2	
	GERAL	Daily Meeting -14h até 14h15	Daily Meeting -14h até 14h15	Daily Meeting -14h até 14h15	Daily Meeting -14h até 14h15	

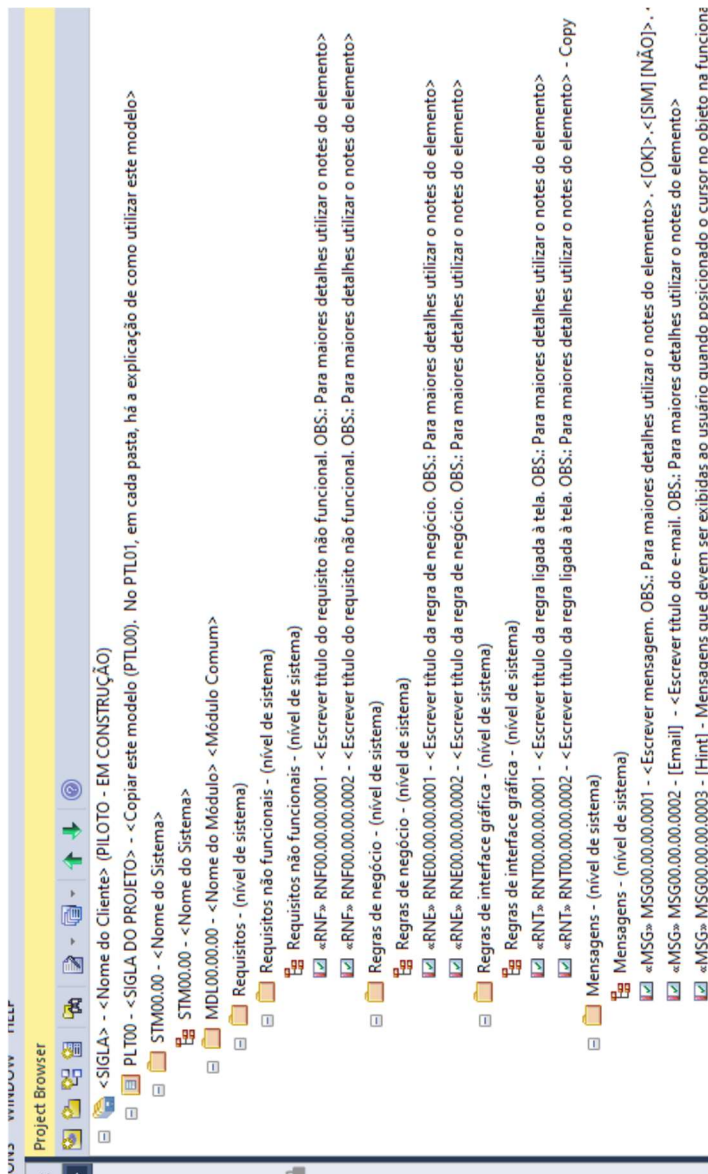
ANEXO F - *Template* de Ata de Reunião

Data:	Local:
Pauta:	
Documentos anexos:	
Participantes: Dia DD/MM/AAAA (H:MM – H:MM) <ul style="list-style-type: none">● Cliente : nomes● Empresa: nomes Apresentação Discussões Encaminhamentos	

ANEXO G - *Template* de Ata de reunião de Sprint

Realizado dia: DD/MM/AAAA	Referente ao <i>Sprint</i> : XX/AAAA Início: DD/MM/AAAA Fim: DD/MM/AAAA
Participantes:	
<ol style="list-style-type: none">1. Objetivo do <i>Sprint</i>2. <i>Review</i> das atividades do <i>Sprint</i> Anterior3. Retrospectiva das atividades do <i>Sprint</i> Anterior:	
<ol style="list-style-type: none">4. Revisão do <i>Backlog</i>5. Tarefas planejadas para o <i>Sprint</i>6. Estimativas7. <i>Sprint Backlog</i>	

ANEXO H - *Template* de produto no Enterprise Architect



ANEXO I - *Template* de documento de Análise Preliminar de Demanda

1. Identificação da demanda

a. Nome da demanda

<Inserir nome da demanda>

b. Dados básicos da demanda

Item	Valor
Cliente	<nome do cliente>
Projeto	<identificação do projeto>
Produto	<identificação do produto em questão>
Responsável	<nome do responsável interno pela demanda>
Responsáveis (Cliente)	<nome dos responsáveis externos pela demanda, bem como, contato>
Data	Selecione a data da demanda
Tipo	Selecione o tipo da demanda
Status	Selecione a situação da demanda

c. Descrição da demanda

<Texto da descrição>

2. Solução atual

a. Descrição da solução atual

<Texto descritivo do cenário + diagramas + imagens + protótipos >

b. Fluxo(s) atual(is)

Abaixo são apresentados os fluxos atuais que estão relacionados com a demanda:

Fluxo A:

- | | |
|----|----------------------------|
| 1. | O usuário clica na opção A |
| 2. | O sistema exibe a tela X |
| 3. | O usuário clica no botão H |
| 4. | O sistema grava as opções |

Fluxo B:

- | | |
|----|----------------------------|
| 1. | O usuário clica na opção A |
| 2. | O sistema exibe a tela X |
| 3. | O usuário clica no botão H |
| 4. | O sistema grava as opções |

3. Dado(s) envolvido(s)

Abaixo são listados os dados envolvidos no cenário desejado:

Dado	Descrição	Tipo
<Dado A>	<Este dado é>	Escolher um item.
		Escolher um item.
		Escolher um item.

4. Solução proposta

a. Descrição da solução proposta

<Texto do cenário desejado + diagramas + imagens + protótipos>

b. Necessidades/Requisitos identificados

Abaixo são apresentados as necessidades expressadas para o atendimento da demanda:

- O sistema/ferramenta deve...
- O sistema/ferramenta deve...
- O sistema/ferramenta deve...

c. Restrições identificadas

Abaixo são apresentadas as restrições identificadas para atendimento da demanda:

- É necessário que..
- O usuário X pode..
- O cadastro Y faz...

d. Fluxo(s) proposto(s)

Abaixo são apresentados os fluxos propostos para a demanda:

Fluxo A:

- | | |
|----|----------------------------|
| 1. | O usuário clica na opção A |
| 2. | O sistema exibe a tela X |
| 3. | O usuário clica no botão H |
| 4. | O sistema grava as opções |

Fluxo B:

- | | |
|----|----------------------------|
| 1. | O usuário clica na opção A |
| 2. | O sistema exibe a tela X |
| 3. | O usuário clica no botão H |

e. Dado(s) envolvido(s)

Abaixo são listados os dados envolvidos na concepção do cenário desejado:

Dado	Descrição	Tipo
<Dado A>	<Este dado é>	Escolher um item.
		Escolher um item.
		Escolher um item.

5. Referencial teórico

Abaixo são apresentados os documentos que podem embasar a execução a demanda:

Nome do documento	Local/Endereço de acesso
<nome físico do documento>	<url, caminho no servidor de arquivos>

6. Glossário de termos e siglas utilizados

Abaixo são listados os termos utilizados na descrição da demanda

Termo/Sigla	Significado
YYYY	<descrição>