



**UNIVERSIDADE
Estadual de LONDRINA**

LETÍCIA MAYUMI DOY OKAMOTO

**ALINHAMENTO DO PROCESSO DE DESENVOLVIMENTO
DE SOFTWARE DO LABORATÓRIO GAIA À
METODOLOGIA ÁGIL SAFE E AOS NÍVEIS G E F DO
MODELO DE QUALIDADE MR-MPS-SW**

LONDRINA-PR

2017

LETÍCIA MAYUMI DOY OKAMOTO

**ALINHAMENTO DO PROCESSO DE DESENVOLVIMENTO
DE SOFTWARE DO LABORATÓRIO GAIA À
METODOLOGIA ÁGIL SAFE E AOS NÍVEIS G E F DO
MODELO DE QUALIDADE MR-MPS-SW**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Rodolfo Miranda de Barros

LONDRINA-PR

2017

Letícia Mayumi Doy Okamoto

Alinhamento do Processo de Desenvolvimento de Software do Laboratório GAIA à metodologia ágil SAFe e aos níveis G e F do modelo de qualidade MR-MPS-SW/ Letícia Mayumi Doy Okamoto. – Londrina-PR, 2017-

84 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Rodolfo Miranda de Barros

– Universidade Estadual de Londrina, 2017.

1. SAFe. 2. MPS.BR. 3. Processo de desenvolvimento de *software*. 4. Projetos ágeis. 5. Qualidade de *software* I. Prof. Dr. Rodolfo Miranda de Barros. II. Universidade Estadual de Londrina. III. Faculdade de Ciência da Computação. IV. Alinhamento do Processo de Desenvolvimento de Software do Laboratório GAIA à metodologia ágil SAFe e aos níveis G e F do modelo de qualidade MR-MPS-SW

CDU 02:141:005.7

LETÍCIA MAYUMI DOY OKAMOTO

**ALINHAMENTO DO PROCESSO DE DESENVOLVIMENTO
DE SOFTWARE DO LABORATÓRIO GAIA À
METODOLOGIA ÁGIL SAFE E AOS NÍVEIS G E F DO
MODELO DE QUALIDADE MR-MPS-SW**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA

Prof. Dr. Rodolfo Miranda de Barros
Universidade Estadual de Londrina
Orientador

Prof. Dr. Segundo Membro da Banca
Universidade/Instituição do Segundo
Membro da Banca

Prof. Dr. Terceiro Membro da Banca
Universidade/Instituição do Terceiro
Membro da Banca

Prof. Ms. Quarto Membro da Banca
Universidade/Instituição do Quarto
Membro da Banca

Londrina-PR, ___ de fevereiro de 2017

Este trabalho é dedicado aos meus pais, que não mediram esforços para me oferecer as melhores oportunidades e sempre me apoiaram em todas as etapas da minha formação

AGRADECIMENTOS

Agradeço primeiramente à minha família, principalmente meus pais Alice e Roberto, por todo suporte, conforto emocional, e ensinamentos que recebi durante todos os momentos da minha vida e que contribuíram para o meu crescimento pessoal.

A todos os professores que tive ao longo da graduação, por toda dedicação em compartilhar seus conhecimentos. Gostaria de agradecer especialmente ao meu orientador Prof. Dr. Rodolfo Miranda de Barros, por guiar a execução deste trabalho, à Prof^a Neyva Maria Lopes Romeiro, por toda orientação e apoio dados durante a iniciação científica, e ao Prof. Dr. Alan Salvany Felinto, por todo auxílio no desenvolvimento do estágio obrigatório.

Também agradeço à todas as pessoas que de alguma forma contribuíram para a minha formação e me auxiliaram durante a graduação e execução deste trabalho.

Por fim, gostaria de agradecer à Universidade Estadual de Londrina e ao Departamento de Computação por toda estrutura que pude usufruir durante esses quatro anos de graduação.

*Feeling a little uncomfortable with your skills is a sign of learning,
and continuous learning is what the tech industry thrives on!
It's important to seek out environments where you are supported,
but where you have the chance to be uncomfortable and learn new things.
(Vanessa Hurst, Co-Founder of Girl Develop It)*

OKAMOTO, L. M. D.. **Alinhamento do Processo de Desenvolvimento de Software do Laboratório GAIA à metodologia ágil SAFe e aos níveis G e F do modelo de qualidade MR-MPS-SW**. 84 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina–PR, 2017.

RESUMO

Atentas ao mercado competitivo, as organizações desenvolvedoras de *software* procuram se organizar de forma a oferecer produtos e serviços de qualidade, desenvolvidos em prazos cada vez menores, além de estarem preparadas para atender às constantes mudanças econômicas do mercado. Para isso, as organizações utilizam um processo de desenvolvimento de *software* (PDS), que precisa ser produtivo, flexível e com qualidade, pois essas características serão refletidas no produto final. Uma maneira de a organização aumentar sua produtividade e rapidez de resposta à mudanças é utilizando uma metodologia ágil, como o SAFe, que possibilita o escalonamento desses princípios à todos os departamentos da empresa. Quanto à qualidade, utiliza-se modelos de referência para mensurá-la, como o MR-MPS-SW, modelo que integra o MPS.BR, programa que busca a melhoria dos processos de *software* brasileiros. Procurando guiar o desenvolvimento de seus produtos, o Laboratório GAIA - Soluções em TIC do Departamento de Computação da UEL, desenvolveu um PDS próprio, que será estudado detalhadamente para que um alinhamento ao SAFe e ao MR-MPS possa ser realizado. Com esse estudo, espera-se encontrar melhorias que possam se aplicadas ao PDS GAIA.

Palavras-chave: Processo de Desenvolvimento de Software, SAFe, MR-MPS-SW, Melhoria do Processo de Software

OKAMOTO, L. M. D.. **Alignment of the GAIA's Laboratory Software Development Process to SAFe agile methodology and G and F levels of MR-MPS-SW quality reference model.** 84 p. Final Project (Bachelor of Science in Computer Science) – State University of Londrina, Londrina-PR, 2017.

ABSTRACT

Attentive to the competitive market, software development organizations seek to organize themselves in order to offer quality products and services, developed in ever shorter deadlines, besides being prepared to respond the market constant economic changes. For this, organizations use a software development process (SDP), which needs to be productive, flexible and with quality, as these characteristics will be reflected in the final product. One way for the organization increase its productivity and responsiveness to changes is to use an agile methodology, such as the SAFe, which allows the scaling of these principles to all the departments of the company. As for quality, reference models are used to measure it, such as MR-MPS-BR, a model that integrates MPS.BR, program that seek to improve brazilian software processes. Looking to guide the development of its products, the GAIA Laboratory - ICT Solutions of UEL Computer Department, developed a SDP of its own, which will be studied in detail so that an alignment to SAFe and MR-MPS can be done. With this study, it is expected to find improvements that can be applied to the GAIA SDP.

Keywords: Software Development Process, SAFe, MR-MPS-SW, Software Process Improvement

LISTA DE ILUSTRAÇÕES

Figura 1 – Ciclo do processo de melhoria de <i>software</i> (Fonte: Sommerville[1])	31
Figura 2 – Configuração Portfolio SAFe. (Fonte: [2])	38
Figura 3 – Exemplo de utilização do Kanban no nível de programa. (Fonte: [2]) . .	45
Figura 4 – Processo de Desenvolvimento de <i>Software</i> GAIA. (Fonte: [3])	58
Figura 5 – Análise Inicial - ANI. (Fonte: [3])	59
Figura 6 – Análise e Planejamento - APL(Parte 1). (Fonte: [3])	60
Figura 7 – Análise e Planejamento - APL(Parte 2). (Fonte: [3])	61
Figura 8 – Execução e Implementação - EXI. (Fonte: [3])	63
Figura 9 – Validação e Testes - VAT. (Fonte: [3])	64
Figura 10 – Entrega - ENT. (Fonte: [3])	65
Figura 11 – Finalização - FIN. (Fonte: [3])	66
Figura 12 – Manter requisitos - MRQ. (Fonte: [3])	67
Figura 13 – Gerenciar Portfólio-GPT. (Fonte: [3])	68
Figura 14 – Gráfico das porcentagens do alinhamento do PDS ao SAFe (Fonte: Autor)	74
Figura 15 – Gráfico das porcentagens do alinhamento do PDS ao MPS (Fonte: Autor)	75
Figura 16 – Processo de Desenvolvimento de <i>Software</i> GAIA modificado. (Fonte: Autor)	77
Figura 17 – Análise de Problemas - APR. (Fonte: Autor)	77
Figura 18 – Análise e Planejamento modificado. (Fonte: Autor)	78
Figura 19 – Execução e Implementação modificado. (Fonte: Autor)	79
Figura 20 – Entrega modificado. (Fonte: Autor)	79

LISTA DE TABELAS

Tabela 1 – Os princípios dos métodos ágeis	29
Tabela 2 – Níveis de maturidade do MR-MPS-SW	52
Tabela 3 – Resultados esperados para o processo de Gerência de Projetos	53
Tabela 4 – Resultados esperados para o processo de Gerência de Requisitos	54
Tabela 5 – Resultados esperados para o processo de Aquisição	54
Tabela 6 – Resultados esperados para o processo de Gerência de Configuração	55
Tabela 7 – Resultados esperados para o processo de Garantia de Qualidade	55
Tabela 8 – Resultados esperados para o processo de Gerência de Portfólio de Projetos	56
Tabela 9 – Resultados esperados para o processo de Medição	56
Tabela 10 – Classificação quanto à implementação dos processos	70
Tabela 11 – Alinhamento do PDS GAIA ao SAFe	72
Tabela 12 – Alinhamento do PDS GAIA ao MR-MPS-SW, parte 1	73
Tabela 13 – Alinhamento do PDS GAIA ao MR-MPS-SW, parte 2	74

LISTA DE ABREVIATURAS E SIGLAS

ADS	Apresentar uma Demo do Sistema
AMP	Avaliação e Melhoria do Processo Organizacional
ANI	Análise Inicial
AP	Atributo de Processo
APL	Análise e Planejamento
AQU	Aquisição
AR	Atividade Rotulada
ART	<i>Agile Release Train</i>
BRE	<i>Backlog Refinement</i>
CL	Classificação
CMMI	<i>Capability Maturity Model Integration</i>
DEV	Desenvolvimento
DFP	Definição do Processo Organizacional
DRE	Desenvolvimento de Requisitos
DRU	Desenvolvimento para Reutilização
DSU	<i>Daily Stand-up</i>
EA	Evidência da Atividade
ENT	Entrega
ERE	Evidência do Resultado Esperado
EXI	Execução e Implementação
FDD	<i>Feature Driven Development</i>
FIN	Finalização
GCO	Gerência de Configuração
GDE	Gerência de Decisões

GFA	Gerenciar Fluxo do ART
GPP	Gerência de Portfólio de Projetos
GPR	Gerência de Projeto
GPT	Gerenciar Portfólio
GQA	Gerência de Qualidade
GRE	Gerência de Requisitos
GRH	Gerência de Recursos Humanos
GRI	Gerência de Riscos
GRU	Gerência de Reutilização
I	Implementado
I&A	<i>Inspect & Adapt</i>
IEC	<i>International Electrotechnical Commission</i>
IEX	<i>Iteration Execution</i>
IPI	<i>Innovation and Planning Iteration</i>
IPL	<i>Iteration Planning</i>
IRT	<i>Iteration Retrospective</i>
IRW	<i>Iteration Review</i>
ISO	<i>International Organization for Standardization</i>
ITP	Integração do Produto
KPI	<i>Key Performance Indicators</i>
MED	Medição
MPS	Melhoria de Processo do <i>Software</i>
MR	Modelo de Referência
MRQ	Manter Requisitos
NBR	Norma Brasileira
NI	Não Implementado

PAI	Parcialmente Implementado
PCP	Projeto e Construção do Produto
PDCA	<i>Plan-Do-Check-Adjust</i>
PDS	Processo de Desenvolvimento de <i>Software</i>
PI	<i>Program Increment</i>
PIP	<i>Program Increment Planning</i>
PMBOK	<i>Project Management Body of Knowledge</i>
PMI	<i>Project Management Institute</i>
PO	<i>Product Owner</i>
POI	Pouco Implementado
PTF	Portfólio
RH	Recursos Humanos
RTE	<i>Release Train Engineer</i>
SAFe	<i>Scaled Agile Framework</i>
SOFTEX	Associação para Promoção da Excelência do Software Brasileiro
SW	<i>Software</i>
TI	Totalmente Implementado
TIC	Tecnologia da Informação e Comunicação
UEL	Universidade Estadual de Londrina
VAL	Validação
VAT	Validação e Testes
VER	Verificação
WBS	<i>Work Breakdown Structure</i>
WIP	<i>Working in Progress</i>
WSJF	<i>Weighted Shortest Job First</i>
XP	<i>Extreme Programming</i>

SUMÁRIO

1	INTRODUÇÃO	25
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Processo de Desenvolvimento de <i>software</i>	27
2.1.1	Desenvolvimento Ágil de <i>Software</i>	28
2.1.2	Qualidade de <i>Software</i>	29
2.2	Melhoria de Processo do <i>Software</i>	30
2.2.1	Medição de processos	31
2.2.2	Análise de processos	32
2.2.3	Mudança de processos	33
2.3	Trabalhos Correlatos	34
3	SCALED AGILE FRAMEWORK - SAFE	37
3.1	<i>Team Level</i>	38
3.2	<i>Program Level</i>	41
3.3	<i>Portfolio Level</i>	46
4	MODELO DE REFERÊNCIA MPS-BR DE SOFTWARE	49
4.0.1	Descrição dos processos	50
4.0.2	Resultados esperados no nível G	53
4.0.3	Resultados esperados no nível F	54
5	PROCESSO DE DESENVOLVIMENTO DE <i>SOFTWARE GAIA</i>	57
5.1	Análise Inicial - ANI	58
5.2	Análise e Planejamento - APL	60
5.3	Execução e Implementação - EXI	63
5.4	Validação e Testes - VAT	64
5.5	Entrega - ENT	65
5.6	Finalização - FIN	66
5.7	Manter Requisitos - MRQ	67
5.8	Gerenciar Portfólio - GPT	67
6	ALINHAMENTO DO PDS GAIA	69
6.1	Metodologia	69
6.1.1	Grau de Rigoriedade	69
6.2	Alinhamento	70
6.2.1	Justificativas	70

6.2.2	Discussão dos Resultados	71
7	CONCLUSÃO	81
	REFERÊNCIAS	83

1 INTRODUÇÃO

Com o crescimento do uso e da criação de *softwares*, faz-se cada vez mais necessário que as empresas que produzem *software* organizem-se de forma a garantir mais produtividade, rapidez e qualidade dos produtos entregues. Tendo isso em mente, uma das opções mais indicadas é a utilização de um processo de desenvolvimento que, se utilizado de maneira correta, pode trazer inúmeros benefícios à organização, melhorando desde a qualidade do *software* produzido até o desempenho das equipes de desenvolvimento.

Buscando oferecer rapidez e dinamicidade ao processo de desenvolvimento de *softwares*, foram desenvolvidos métodos denominados ágeis, que valorizam a interação constante com cliente e a entrega contínua e incremental de funcionalidades desenvolvidas em ciclos iterativos[1]. Inicialmente, a metodologia foi desenvolvida para ser aplicada apenas no setor de desenvolvimento, mas sua aplicabilidade em outros setores da empresa foi percebida, como mostra o SAFe. O *Scaled Agile Framework*, SAFe, é uma metodologia ágil que pode ser estendida para abranger equipes de desenvolvimento que envolvem muitas pessoas ou para outros setores, podendo ser escalonada até abranger toda a empresa[2].

Quanto à qualidade, ela pode ser quantificada através da utilização de modelos de referência, como o MPS.BR. A partir dessa análise, o modelo de referência pode oferecer guias de como melhorias podem ser implementadas no processo, garantindo uma maior qualidade no produto final[4].

O MPS.BR é um programa que busca promover melhorias no processo de *software* brasileiro. É composto por cinco componentes, sendo um deles o MR-MPS-SW, Modelo de Referência MPS de *Software*, que baseado em normas nacionais, NBR ISO/IEC 12207, internacionais, ISO/IEC 33020, e no modelo CMMI-DEV[®], procura quantificar a qualidade que se espera de um *software* e prover ferramentas para que esse atinja bons patamares[5].

Identificando a importância de ter um processo de desenvolvimento que proporcione produtividade e qualidade ao produto final, a GAIA possui seu próprio processo de desenvolvimento de *software*[3]. Procurando melhorar a produtividade e a qualidade do processo, uma análise sobre o alinhamento deste a uma metodologia ágil e a um modelo de qualidade pode ser feita.

Portanto, a finalidade deste trabalho é analisar o quão alinhado está o Processo de desenvolvimento de *software* da GAIA ao SAFe e aos níveis G e F do MR-MPS-SW, mapeando quais processos e atividades estão sendo implementados pelo PDS. Com esses dados, será possível identificar melhorias que precisam ser feitas para que o PDS alcance maiores índices de produtividade, flexibilidade à mudanças, qualidade do processo e qualidade agregada ao produto.

Este trabalho é organizado de forma que o capítulo 2 apresenta a fundamentação teórica na qual o trabalho se baseia, o capítulo 3 traz detalhes sobre o *framework* SAFe e descreve as principais atividades que o compõe, o capítulo 4 apresenta o funcionamento do MR-MPS-BR e os resultados esperados para os níveis de maturidade, o capítulo 5 detalha as etapas do Processo de Desenvolvimento de *Software* da GAIA, o capítulo 6 traz o alinhamento do PDS GAIA aos modelos propostos e discussão dos resultados obtidos, e o capítulo 7 apresenta as considerações finais sobre o trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Processo de Desenvolvimento de *software*

Segundo Pressman[4], um processo é um conjunto de atividades, ações e tarefas realizadas na criação de algum produto de trabalho. No contexto da engenharia de *software*, um processo é uma abordagem adaptável cuja intenção é sempre entregar um *software* dentro do prazo e com qualidade suficiente para satisfazer os clientes, aqueles que patrocinaram sua criação, e os futuros usuários[4]. Ainda de acordo com Pressman[4], que utiliza o termo metodologia (*framework*) de processo, um PDS genérico compreende cinco atividades:

1. Comunicação: é de vital importância para o projeto que haja uma boa comunicação entre seus realizadores, os clientes e demais interessados. Com isso, busca-se a total compreensão dos objetivos a serem alcançados pelo projeto;
2. Planejamento: envolve o desenvolvimento de um plano para guiar o processo de desenvolvimento. Descreve tarefas a serem feitas, riscos, recursos necessários e o cronograma de trabalho;
3. Modelagem: criam-se modelos que possibilitem a visualização de todo o *software*, buscando o projeto que melhor atende às suas necessidades;
4. Construção: implementação e testes são feitos;
5. Emprego: é entregue para avaliação do cliente o *software* completo (ou um incremento parcialmente efetivado), esperando-se um *feedback*.

Essas atividades são complementadas por atividades de apoio, que auxiliam a equipe a gerenciar, controlar o progresso, a qualidade, as mudanças e os riscos. Os processos podem ser adaptados independentemente do tamanho ou complexidade do *software*, essa adaptação também pode ser para uma necessidade específica de um projeto. A maioria das empresas prefere desenvolver os próprios processos, mas há inúmeros PDSs disponíveis para uso que geralmente são categorizados em:

- Dirigido a plano ou Prescritivo: abordagem que foca no detalhamento da definição, identificação e aplicação de atividades e tarefas do processo[4], no planejamento com antecedência e na avaliação rigorosa através da comparação do progresso com o planejamento inicial[1];

- Ágeis: ressaltam a agilidade do projeto, através de princípios que pregam mais informalidade, flexibilidade e planejamento gradativo.

2.1.1 Desenvolvimento Ágil de *Software*

O ambiente de desenvolvimento de software nos dias de hoje sofre com mudanças rápidas, precisando responder a novas oportunidades, a mudanças econômicas e ao surgimento de novos concorrentes. Assim, o desenvolvimento e entrega rápidos são requisitos críticos no desenvolvimento de sistemas de *software*[1]. PDSs prescritivos, que especificam completamente os requisitos antes de planejar, construir e testar o *software* não estão adaptados para atender essa nova realidade. Por isso, em 2011, Kent Beck e outros dezesseis desenvolvedores e consultores da área de *software* assinaram o “Manifesto para o Desenvolvimento Ágil de Software” (“*Manifesto for Agile Software Development*”), e passaram a valorizar mais[6]:

- os indivíduos e as interações ao invés dos processos e ferramentas;
- o *software* funcionando ao invés de uma documentação abrangente;
- a colaboração com o cliente ao invés da negociação de contratos;
- responder a mudanças ao invés de seguir um plano

Os processos de desenvolvimento ágil desenvolvem o *software* em uma série de pequenos incrementos, sendo que cada um inclui uma nova funcionalidade ao sistema. Essas versões são disponibilizadas aos clientes ao final de cada iteração, assim, um rápido *feedback* sobre a evolução dos requisitos é obtido. Uma iteração é o período de tempo disponível para planejar, desenvolver e testar um incremento, sua duração é adaptada em cada organização, podendo durar semanas ou até meses.

Os diversos PDSs ágeis existentes, como o *Scrum* e o *XP*, compartilham um conjunto de princípios baseados no Manifesto Ágil. Esses princípios são apresentados na Tabela 1.

Tabela 1 – Os princípios dos métodos ágeis

Princípios	Descrição
Envolvimento do cliente	Os clientes devem estar intimamente envolvidos no processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos do sistema e avaliar suas iterações.
Entrega incremental	O <i>software</i> é desenvolvido em incrementos com o cliente especificando os requisitos para serem incluídos em cada um.
Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Membros da equipe devem desenvolver suas próprias maneiras de trabalhar, sem processos prescritivos.
Aceitar as mudanças	Deve-se ter em mente que os requisitos do sistema vão mudar. Por isso, projete o sistema de maneira a acomodar essas mudanças.
Manter a simplicidade	Focalize a simplicidade, tanto do <i>software</i> a ser desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.

Fonte: [Sommerville\[1\]](#)

2.1.2 Qualidade de *Software*

O termo qualidade de software pode ser definido, de forma geral, como uma gestão de qualidade efetiva cuja aplicação criará um produto útil que fornecerá valor mensurável para aqueles que o produziram e para aqueles que o utilizarão[4].

Essa definição foca nas expectativas de três visões:

- Visão do usuário: interessado na utilização e desempenho do *software*, o cliente espera que as funções e recursos desejados estejam presentes e ofereçam confiabilidade e isenção de falhas;
- Visão do desenvolvedor: busca atender as especificações propostas pelo cliente e preocupa-se com medidas internas, como um baixo tempo de resposta e a fácil manutenção;
- Visão do gerente de desenvolvimento: avalia custos, prazos, conformidade e outros critérios baseados nos interesses da empresa.

É possível avaliar a qualidade a partir de padrões, como o ISO 9126, padrão internacional desenvolvido para identificar os atributos fundamentais de qualidade de um *software* para computador. As características e sub-características avaliadas por esses padrões são[7]:

- Funcionalidade: conjunto de funções que satisfazem necessidades implícitas e explícitas conforme indicado na adequação, acurácia, interoperabilidade, conformidade e segurança;
- Confiabilidade: capacidade do *software* em permanecer disponível para uso ao longo de um período de tempo. Seus sub-atributos são maturidade, tolerância a falhas e recuperabilidade;
- Usabilidade: grau de facilidade de uso do *software* pronto, envolvendo compreensibilidade, facilidade de aprendizado e operabilidade;
- Eficiência: relação entre desempenho e recursos utilizados pelo *software*. Indicado pelo comportamento em relação ao tempo e ao uso de recursos;
- Manutenibilidade: Facilidade com a qual correções e alterações podem ser feitas, conforme indicado pela analisabilidade, modificabilidade, estabilidade e testabilidade;
- Portabilidade: Facilidade com a qual um *software* é portado para outro ambiente. Envolve adaptabilidade, facilidade de instalação, conformidade e facilidade de substituição.

Ao final de um projeto, espera-se um *software* com qualidade e que contenha essas características e satisfaça essas visões. Para isso, o processo de desenvolvimento deve sofrer melhorias para que consiga refletir sua qualidade no produto final.

2.2 Melhoria de Processo do *Software*

Devido ao cenário competitivo em que se encontra o mercado de desenvolvimento de *softwares*, as empresas procuram oferecer produtos com qualidade, baixo custo e desenvolvidos dentro do prazo de entrega, que tendem a ficar cada vez mais curtos. Por isso, muitas organizações procuram melhorar seus processos de desenvolvimento através de mudanças que possam melhorar alguns atributos do processo, como aumentar sua qualidade e reduzir seus custos e tempo de desenvolvimento, benefícios que são refletidos no produto final.

De acordo com Pressman[4], um processo de Melhoria de Processo de *Software* (MPS) define características que devem estar presentes em um processo de *software* eficaz, um método para determinar se tais características estão presentes, um mecanismo para organizar os resultados das avaliações e uma estratégia para ajudar a organização a implementar as características faltantes ou parcialmente implementadas.

A junção das definições dessas características, seus métodos de avaliação e implantação, formam o conceito de maturidade do processo de *software* de uma organização, um

indicador métrico da qualidade que busca classificar em um nível o processo de desenvolvimento, a fim de oferecer estratégias para que este supra as características que não foram implementadas e melhore sua qualidade. Para avaliar a maturidade de um processo, são utilizados Modelos de Maturidade, ou Modelos de Referência, que definem níveis e características esperadas para cada processo[4].

Alguns exemplos de normas e modelos de referência utilizados no processo de melhoria de processo de *software* são o CMMI-DEV (*Capability Maturity Model Integration for Development*[8]), as normas internacionais ISO/IEC 12207 (Engenharia de Sistemas de *Software* - Processo de Ciclo de Vida de *Software*[9]) e ISO/IEC 15504 (Tecnologia da Informação - Avaliação de Processos[10]), e o MR-MPS-SW (Modelo de Referência para Melhoria de Processo de *Software*[5]).

O processo de melhoria de *software* ainda pode ser definido como um processo cíclico, mostrado na Figura 1, formado por três subprocessos descritos nas próximas subseções.

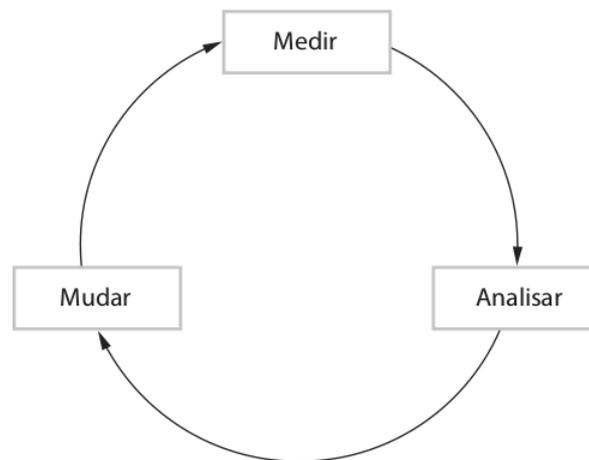


Figura 1 – Ciclo do processo de melhoria de *software* (Fonte: Sommerville[1])

2.2.1 Medição de processos

A medição de processos é responsável por criar dados quantitativos sobre o processo de *software*, que serão utilizados para avaliar a melhoria da eficiência de um processo. A obtenção desses dados é baseada em métricas, os três tipos principais são[1]:

1. Tempo necessário para conclusão de um processo específico: Pode ser o tempo total dedicado para o processo, o tempo de calendário, o tempo dispendido no processo por engenheiros específicos, entre outros;

2. Os recursos necessários para um determinado processo: podem envolver os gastos com recursos computacionais, gastos relacionados com a conclusão de determinado processo, a relação entre quantidade de pessoas/dia;
3. O número de ocorrências de um determinado evento: é a contagem da quantidade de vezes em que um evento ocorre. Estes eventos podem ser: mudanças de requisitos, descoberta de falha/erro, entre outros.

A partir das informações quantitativas é possível inferir, por simples comparações, o nível do processo e suas possíveis melhorias. No entanto, essas evidências precisam ser interpretadas em conjunto com informações qualitativas das mudanças, para que seja possível identificar se as mudanças nos resultados se devem exclusivamente à mudanças relacionadas diretamente ao processo.

2.2.2 Análise de processos

Os estudos necessários para compreender as principais características dos processos e como estes são executados na prática, fazem parte da análise de processo. Apesar da Figura 1 sugerir que a medição de processos é seguida pela análise de processos, essas atividades ocorrem de maneira intercalada, já que é preciso realizar algumas análises para saber o que medir, e enquanto realiza-se as medições, inevitavelmente, uma compreensão mais profunda do processo envolvido será desenvolvida[1].

Os objetivos gerais da análise de processos são:

1. Entender as atividades envolvidas no processo e como se relacionam;
2. Entender os relacionamentos entre as atividades de processos e as medições;
3. Comparar os processos analisados com outros processos compatíveis localizados em outro lugar da organização ou com processos idealizados do mesmo tipo.

Alguns aspectos que podem ser investigados durante a análise de processos para alcançar os objetivos expostos são:

- Adoção e padronização: o processo deve estar documentado e padronizado em toda a organização para que as medições sejam válidas e comparáveis;
- Prática de engenharia de *software*: as boas práticas de engenharia de *software* devem estar incluídas no processo;
- Restrições organizacionais: verificar se existem restrições organizacionais que afetem o processo e impossibilitem que mudanças sejam realizadas;

- Comunicações: analisar como ocorre o gerenciamento de comunicações no processo;
- Introspecção: os atores envolvidos no processo devem pensar e discutir, além de ter mecanismos, para melhorar o processo;
- Aprendizagem: as pessoas devem aprender sobre os processos através de manuais e programas de treinamento;
- Suporte a ferramentas: analisar quais aspectos do processo possuem uma ferramenta de suporte e quais não, averiguar quais ferramentas são eficazes e eficientes, pensar em novas ferramentas que possam melhorar o acompanhamento dos processos.

Ao final de uma análise de processo de *software*, espera-se obter uma compreensão mais profunda do processo e de melhorias futuras que podem ser feitas[1].

2.2.3 Mudança de processos

A etapa de mudança de processo realiza as modificações nos processos. Essas mudanças podem ocorrer por meio da introdução de novas práticas, métodos ou ferramentas, da troca da ordem das atividades do processo, da introdução de novos papéis e responsabilidades, entre outros[1]. Existem cinco estágios principais para realização das mudanças:

1. Identificação de melhorias: relacionado com os resultados da análise de processo, propõe melhorias possíveis de serem aplicadas;
2. Priorização de melhorias: quando existem muitas mudanças a serem feitas, é preciso elencar ordenadamente as mudanças, pois não é possível realizar todas simultaneamente;
3. Introdução de mudanças de processos: integrar as mudanças às demais atividades do processo;
4. Treinamento de processos: realização de treinamentos para que as pessoas possam entender as mudanças e para que essas sejam essencialmente positivas;
5. Ajuste de mudanças: deve-se ficar atento à pequenos problemas que impeçam as mudanças de funcionarem como planejado, realizando ajustes para que possam ser utilizadas da melhor forma.

Ao realizar a etapa de mudanças de processos completa-se um ciclo do processo de melhoria de processos de *software*, como mostrado na Figura 1. Novas melhorias podem ser realizadas voltando ao início do esquema.

2.3 Trabalhos Correlatos

O uso de modelos de referência, como o MPS.BR e o CMMI, para mensurar a maturidade de processos de desenvolvimento de *software* é uma prática comum. Em [11], estuda-se a aplicação das normas ISO 9001:2000 e dos modelos de maturidade CMMI e MPS.BR para alcançar altos índices de maturidade no PDS da empresa BL Informática. Como resultado, problemas comuns, como atrasos, extrapolação de orçamentos e baixa especificação dos requisitos, foram minimizados. Houve também o crescimento da organização, passando de 6 pessoas trabalhando nos projetos de *software* para 30, em apenas três anos.

Também é possível analisar os níveis de maturidade de metodologias ágeis. Em [12], analisa-se a viabilidade da utilização das metodologias ágeis FDD e Scrum por empresas que buscam utilizar o MPS.BR como modelo de qualidade, concluindo que nenhuma das duas metodologias satisfazem os níveis A e B do MPS.BR, sugerindo que suas adoções não seriam adequadas para empresas que queiram galgar todos os níveis de maturidade. Em [13] a integração da metodologia ágil XP ao nível G do MPS-BR foi analisada e classificada, apresentando gráficos sobre o atendimento dos resultados esperados.

Quanto ao SAFe, os trabalhos encontrados discutem a viabilidade de sua aplicação e as melhorias alcançadas com ela. [14], [15] e [16] apresentam casos de empresas de grande porte que adotaram o SAFe como método para escalonar o uso de metodologia ágil. No primeiro caso, foram estudadas as formas de aplicação do método em duas linhas de negócio cujos colaboradores se encontravam distribuídos em diferentes países, concluindo que a melhor aplicação foi aquela que ofereceu treinamento com antecedência, manteve as pessoas informadas e engajadas, contratou um *expert* para auxiliar no treinamento, houve uma boa preparação para as reuniões de planejamento para o PI (*Program Increment*) e manteve um RTE (*Release Train Engineer*) o tempo todo.

Os trabalhos [15] e [16] apresentaram casos de organizações que empregaram o SAFe em apenas uma parte de todo o processo de desenvolvimento. Em ambos os casos, houveram aumentos na produtividade e dependências entre times foram resolvidas mais rapidamente, o alinhamento entre os times também foi melhorado.

Outro trabalho utilizado como referência é [17], em que o alinhamento dos processos de desenvolvimento de *software* do Laboratório GAIA aos níveis G e F do MPS-BR foram analisados com o objetivo de melhorá-los.

Estes trabalhos apresentaram a análise da adesão de metodologias e processos de desenvolvimento à modelos de maturidade e ao SAFe, e concluíram, de forma geral, que melhorias ocorreram em diversas questões dentro das organizações. Dessa forma, o estudo do alinhamento do PDS GAIA tanto ao modelo de maturidade do MPS.BR quanto ao SAFe, trará a análise de diferentes aspectos que podem ser melhorados no

processo de desenvolvimento, como produtividade e eventuais falhas de comunicação. Ao final deste trabalho, serão sugeridas melhorias que podem ser implantadas no PDS GAIA para aumentar a adesão ao modelo de referência e à metodologia ágil estudados.

3 SCALED AGILE FRAMEWORK - SAFE

Com a evolução do cenário de desenvolvimento de *software* e a demanda por aplicações maiores, mais sofisticadas e mais complexas, houve o incentivo para formação de várias equipes com dezenas, centenas de desenvolvedores, tornando difícil gerenciar suas funções, tarefas, prazos de entrega, alinhamento e sincronia. Além disso, empresas de grande porte encontraram dificuldades para implantação da metodologia ágil, devido à resistência cultural, falta de experiência dos gerentes de projeto com a metodologia e a dificuldade de interação dos diversos departamentos da empresa.

Nesse sentido, Dean Leffingwell criou o SAFe [2], acrônimo para *Scaled Agile Framework*, uma metodologia ágil que pode ser estendida para abranger equipes maiores e outros departamentos da empresa, podendo ser escalonada até abranger toda a organização. A metodologia tem como base o *Lean Thinking* e outras duas metodologias ágeis, o Scrum e o XP (*Extreme Programming*), além de possuir uma abordagem bem documentada para sua aplicação.

O *Lean Thinking* é uma filosofia e estratégia de negócios que busca aumentar a satisfação dos clientes através do melhor uso dos recursos. Acredita-se que entendendo o que é o valor para o cliente, será possível eliminar desperdícios, provocando um melhoramento contínuo nos processos de produção e alavancando a competitividade da empresa. Assim, o SAFe combina os conhecimentos das metodologias Ágeis no desenvolvimento de sistemas com a estratégia de desenvolvimento de produtos do Lean, fornecendo melhorias na produtividade, na qualidade e no engajamento dos funcionários.

O SAFe atualmente encontra-se na versão 4.5 e apresenta quatro configurações: *Essential SAFe*, *Large Solution SAFe*, *Portfolio SAFe* e *Full SAFe*. Neste trabalho será utilizado a configuração *Portfolio SAFe*, como mostrado na Figura 2, indicado para empresas que desenvolvem múltiplas soluções, o que requer um gerenciamento de portfólio para coordenação, estratégia e investimento dos projetos.

A configuração utilizada possui 3 níveis: *Portfolio*, *Program* e *Team*. Cada nível do SAFe fornece abstrações organizacionais com papéis e práticas voltadas a orientar a agilidade em grandes empresas de desenvolvimento de software.

Para uma melhor compreensão do alinhamento a ser realizado entre o SAFe e o PDS GAIA, as atividades do *framework* serão rotuladas e descritas nas próximas seções[2].

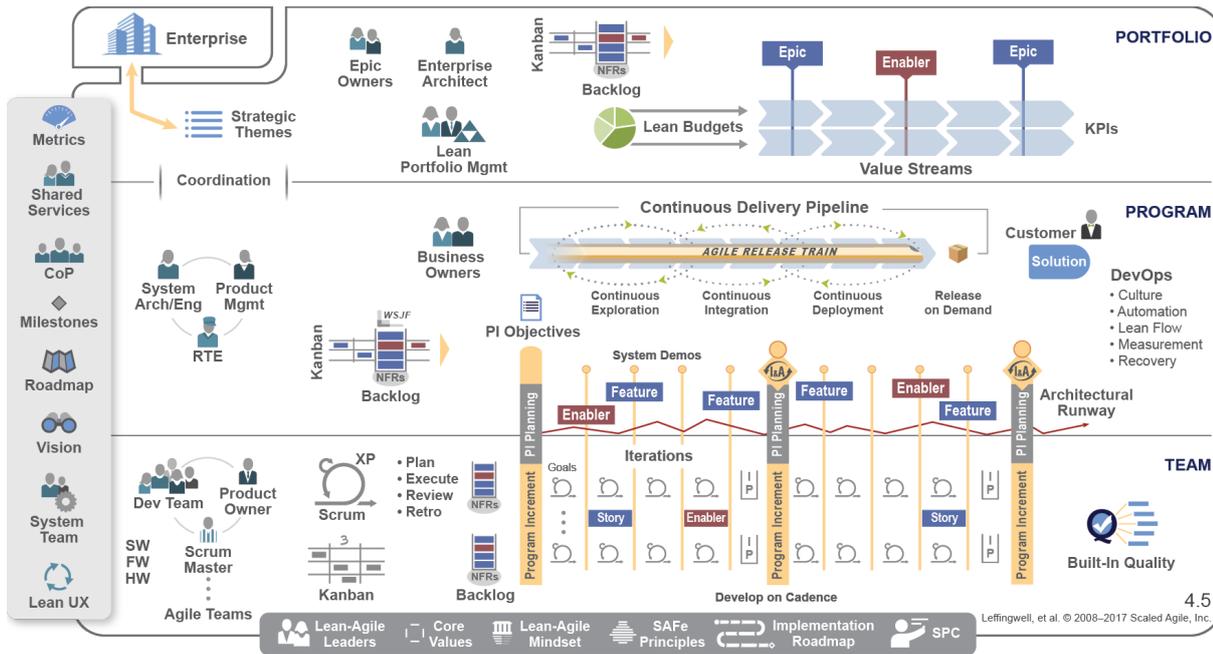


Figura 2 – Configuração Portfólio SAFe. (Fonte: [2])

3.1 Team Level

Este nível fornece um modelo organizacional dos artefatos, papéis, e processos necessários para que as equipes ágeis realizem suas atividades. Cada equipe escolhe sua metodologia, ScrumXP ou Kanban, e seus membros, que variam entre 5 e 9, se dividem nos papéis:

- Time de desenvolvimento: composto por desenvolvedores, testadores e outros especialistas que trabalham de forma colaborativa para entregar uma funcionalidade;
- *Product Owner (PO)*: encarregado do *backlog* do time, organiza a lista de Histórias de Usuário, as tarefas de manutenção e refatoração;
- *Scrum Master*: funciona como um treinador da equipe ajudando os membros a passarem por impedimentos e promovendo um bom ambiente para o alto desempenho da equipe.

Assim, cada time é capaz de planejar, desenvolver, testar e ajustar uma funcionalidade a cada iteração ou *Sprint*, realizando o ciclo PDCA (*Plan-Do-Check-Adjust*) e a entrega da funcionalidade ao final desta. Cada iteração possui os seguintes eventos, alguns destes requerem mais de uma atividade:

- *Iteration Planning*: reunião em que a equipe decide quais os objetivos e quais histórias, tarefas, serão entregues nessa *Sprint*:

- Estabelecer Velocidade (IPL01): A equipe quantifica a sua capacidade de realizar o trabalho nesta iteração. Cada membro avalia sua disponibilidade e tempo de aprendizado. Também é levado em conta o tempo de manutenção e outras atividades independentes;
 - Análise e Estimativa de Histórias (IPL02): Cada caso de uso do *backlog* do time é discutido e suas dificuldades, complexidade e critérios de aceitação são avaliados, chegando a uma estimativa do tamanho da história. Geralmente, o *backlog* também possui atividades de arquitetura e infraestrutura, que são avaliadas similarmente;
 - Elaborar Tarefas (IPL03): esta é uma atividade opcional, em que as equipes fragmentam os casos de uso em tarefas e as dividem entre si. Esta atividade é comum de ser feita em equipes novas, que ainda estão compreendendo sua capacidade e velocidade;
 - Estabelecer os Objetivos da Iteração (IPL04): após a compreensão do *backlog*, a equipe sintetiza um ou mais objetivos para a iteração, baseando-se nos objetivos passados pelo *Program Level*;
 - Obter Comprometimento (IPL05): Os membros da equipe e o *product owner* concordam com as histórias a serem trabalhadas e os objetivos da iteração. Toda a equipe se compromete com os objetivos da iteração, e o escopo permanece o mesmo durante todo o ciclo;
- *Iteration Execution*: é quando a equipe desenvolve um incremento efetivo, de alta qualidade, funcionando e testado dentro do prazo de entrega. O foco do evento é entregar as histórias com as quais a equipe se comprometeu na reunião de planejamento e atender os objetivos da iteração:
 - Acompanhar o Progresso da Iteração (IEX01): o acompanhamento requer visibilidade do atual estado dos casos de uso, testes, correções e outras atividades nas quais a equipe está trabalhando durante a iteração. As equipes que seguem o Kanban utilizam os seus quadros (*Kanban boards*), e as que seguem o Scrum, um *story board*;
 - DSU: *The Daily Stand-up* (IEX02): reuniões diárias com 15 minutos de duração em que a equipe coordena o trabalho respondendo as perguntas: O que fiz ontem para avançar em direção aos objetivos da iteração?, O que conseguirei fazer hoje para avançar em direção aos objetivos da iteração? e O que está nos impedindo de avançar em direção aos objetivos da iteração?. Essas reuniões têm como objetivo identificar problemas e dependências, e manter uma comunicação constante;

- Gereciar o *Working-In-Progress* (IEX03): Limitar o WIP é uma estratégia para evitar gargalos no desenvolvimento, ajudando a melhorar o fluxo de trabalho, o foco, o compartilhamento de informações e a ideia de propriedade coletiva. Esse controle força que a taxa de entrada, de histórias e objetivos, seja compatível com a capacidade do time;
 - Construir com Qualidade (IEX04): o SAFe prescreve um conjunto de cinco práticas de qualidade e engenharia que contribuem para agregar qualidade às funcionalidades: *Test-First* - derivada do XP (*eXtreme Programming*) esta prática planeja os testes antes da implementação, assim as entregas são feitas focadas nos resultados, Integração contínua, Refatoração, Trabalho em pares e propriedade coletiva. Incorporar qualidade desde o início do trabalho faz com que as entregas sejam mais rápidas, fáceis e menos onerosas;
 - Aprovação Contínua de Histórias (IEX05): aceitar que histórias foram completamente atendidas melhora continuamente o fluxo de trabalho. As equipes demonstram suas funcionalidades prontas antes de terminar a iteração, assim podem abordar problemas de forma rápida e eficiente, retrabalhando seus casos de uso sem trocar de contexto;
 - Automação de Testes (IEX06): sempre que possível, os critérios para o comportamento adequado do sistema, definidos pelo *Product Owner* e as equipes, são transformados em testes para avaliação de casos de uso que podem ser executados repetidamente para garantir a adequação ao uso e a conformidade contínua do sistema à medida que ele evolui;
-
- *Backlog Refinement* (BRE01): ocorre uma ou duas vezes durante a iteração com o objetivo de refinar, analisar e estimar histórias de usuários e seus requisitos, no *backlog* do time;
 - *Iteration Review* (IRW01): reunião realizada ao final da iteração, a equipe mostra o trabalho feito ao *Product Owner* e demais interessados, que analisam se o resultado é o desejado e oferecem um *feedback*. Com o retorno, a equipe discute os porquês de não conseguirem completar suas tarefas, chegando à conclusão, geralmente, de que houve a descoberta de novos riscos, as prioridades foram modificadas, as estimativas estavam imprecisas ou houve excesso de compromissos. Esses problemas encontrados servirão para melhorar o planejamento das próximas iterações;
 - *Iteration Retrospective* (IRT01): ao final de uma iteração, os membros da equipe analisam se os objetivos propostos para a iteração foram atingidos e o que pode ser melhorado para a próxima. A avaliação é baseada nas informações qualitativas e quantitativas apresentadas durante a *Iteration Review*;

- *Innovation and Planning (IP) Iteration (IPI01)*: é uma iteração que oferece às equipes a chance de explorar e inovar, obter novos conhecimentos e técnicas, trabalhar focado na parte técnica do sistema ou, como nenhuma funcionalidade está prevista para ser implementada nessa iteração, trabalhar as tarefas atrasadas devido a dependências e imprevistos. Caso esteja próximo de realizar a entrega de uma versão, as equipes podem utilizar esse *sprint* para realizar testes, verificação, validação e documentações finais do sistema.

As várias equipes utilizam esses eventos para administrar e sincronizar o andamento de cada iteração, cuja duração é a mesma para todas. Para garantir que as funcionalidades criadas pelas equipes funcionam juntas, uma versão com todo o sistema integrado é entregue para avaliação, como descreve a atividade ADS01 do *Program Level*.

3.2 *Program Level*

O nível de programa é onde as equipes de desenvolvimento, os *stakeholders* e os recursos estão dedicados ao desenvolvimento de soluções em andamento que precisam ser entregues, sejam elas produtos, serviços ou sistemas, internos ou externos à empresa. Para desenvolver essas soluções, as equipes, papéis e atividades desse nível são organizados em torno da metáfora do *Agile Release Train (ART)*, que como um trem, tem cronograma de partida e chegada, velocidade padrão e planejamento previsível. Assim, os ARTs são organizações virtuais auto gerenciáveis, formadas por equipes ágeis que planejam, executam e entregam juntas em um fluxo contínuo. Cada ART é composto por 5 a 12 equipes ágeis, envolvendo entre 50 a 125 pessoas. Dependendo da complexidade da solução, um ou mais ARTs serão necessários para o seu desenvolvimento.

Os papéis do *Program Level* são voltados a guiar e direcionar o andamento do ART, ajudando a alinhar as equipes de desenvolvimento a um objetivo comum:

- *Arquiteto/Engenheiro de Sistema (System Architect/Engineer)*: é um indivíduo ou uma pequena equipe que define a estrutura geral do sistema, determina os principais elementos e subsistemas, também ajuda a definir os requisitos não funcionais, as interfaces e colaborações entre elas;
- *Product Management*: é a voz do cliente, trabalha junto com os *Product Owners* das equipes para entender as necessidades do cliente e garantir seu atendimento, definindo recursos e participando da validação. São responsáveis pelo *backlog* do programa;
- *Release Train Engineer (RTE)*: atua como o *Scrum Master* para ART, otimizando o fluxo e garantindo que o trem ande dentro dos trilhos;

- *Business Owners*: é um pequeno grupo de *stakeholders* que têm a responsabilidade de garantir a adequação, governança, e o retorno do investimento de uma solução desenvolvida pelo ART.

O *Backlog* do programa possui as definições das aplicações, *features* e *enablers* previstos para desenvolver um ART que atenda as necessidades do cliente e forneça uma boa arquitetura ao sistema, respectivamente. Mantê-lo atualizado e organizado é essencial para o bom planejamento e andamento do desenvolvimento do ART. Dessa forma, a melhor maneira de organizar o *backlog* é priorizar algumas tarefas utilizando método *Weighted Shortest Job First* (WSJF), que possui como operadores o valor retornado pelo desenvolvimento dessa tarefa ao usuário e ao negócio (*User|Business Value*), a criticidade de sua entrega em relação ao tempo (*Time Criticality*), os riscos reduzidos pelo seu desenvolvimento ou se gerará uma nova oportunidade (*Risk Reduction | Opportunity Enablement Value*) e o seu tamanho (*Job Size*), medido em relação à quantidade de histórias que a compõe. Assim, segue-se a fórmula 3.1[2].

$$WSJF = \frac{User|Business\ Value + Time\ Criticality + Risk\ Reduction|Opportunity\ Enablement\ Value}{Job\ Size} \quad (3.1)$$

Para controlar o andamento do ART, utiliza-se o conceito de *Program Increment* (PI), período de duração fixa em que será desenvolvida uma parte maior (valor incremental) do *software* ou sistema, envolvendo pequenas funcionalidades desenvolvidas pelas várias equipes. Um PI tem duração de oito a doze semanas, e comumente envolve quatro iterações de desenvolvimento e uma de *Innovation and Planning* (IP), etapa em que ocorre também o planejamento da próxima PI.

Os principais eventos e atividades que auxiliam na coordenação do ART são:

- *PI Planning*: Todos os membros dos times e demais interessados, realizam uma reunião, conduzida pelo RTE, para alinhar todas as equipes com o objetivo comum do PI:
 - Contextualizar o Negócio (PIP01): um executivo/proprietário descreve o atual estado do negócio e apresenta uma perspectiva sobre o quão bem as soluções atuais estão atendendo as necessidades dos clientes;
 - Apresentar a Visão de Produto (PIP02): o *Product Management* apresenta a visão de produto para as próximas 10 principais funcionalidades, geralmente;
 - Apresentar Visão de Arquitetura e Práticas de Desenvolvimento (PIP03): O arquiteto/engenheiro de sistema apresenta a visão da arquitetura. Além disso, um desenvolvedor sênior pode apresentar as práticas propostas por metodologias ágeis que estarão sendo implantadas no próximo PI;

- Apresentar o Contexto de Planejamento (PIP04): o RTE apresenta o processo de planejamento e os resultados esperados da reunião;
 - Reunião 1 das equipes (PIP05): as equipes estimam suas capacidades para cada iteração e analisam quais itens do *backlog* serão necessários para realizarem as funcionalidades atribuídas a elas. Também desenvolvem seus planos preliminares, iteração por iteração, apontando riscos e dependências com tarefas de outras equipes;
 - Revisar os Planos Preliminares (PIP06): as equipes apresentam os objetivos, riscos potenciais e dependências de seus planos preliminares. *Business Owners, Product Management, stakeholders* e demais equipes analisam e fornecem informações;
 - Revisar o Gerenciamento e Solucionar Problemas (PIP07): é provável que os planos apresentados tenham desafios como o escopo, restrição de recursos e dependências, então o *Product Management* negocia o escopo e tenta solucionar tais problemas, concordando com vários ajustes;
 - Comunicar Ajustes no Planejamento (PIP08): o gerente descreve as mudanças feitas no planejamento do escopo e dos recursos;
 - Reunião 2 das Equipes (PIP09): as equipes continuam seu planejamento, realizando os ajustes necessários. Ao finalizarem, os *business owners* atribuem valores de negócio aos objetivos, estimando o impacto corporativo destes sobre a estratégia da empresa e suas prioridades;
 - Revisão Final dos planos (PIP10): as equipes apresentam seus planos para todos e declaram seus objetivos, riscos e impedimentos. Se os planos forem aprovados pelos clientes, agregam-se os objetivos das equipes, sintetizando o objetivo do PI;
 - Identificar Riscos (PIP11): durante o planejamento, as equipes identificaram riscos e impedimentos que poderiam afetar suas capacidades de atingirem seus objetivos. Estes riscos são abordados de forma clara e discutidos;
 - Votação na Confiança (PIP12): as equipes votam na confiança que têm para cumprir seus objetivos e os do PI;
 - Refazer Plano (PIP13): se necessário, as equipes refazem seus planos, até atingirem um alto grau de confiança;
 - Realizar retrospectiva de planejamento (PIP14): ao final da reunião, o RTE avalia o que ocorreu bem ou não e o que pode ser melhorado.
- Apresentar uma Demo do Sistema (ADS01): realizada ao final de cada iteração, fornece uma visão integrada entre todas as funcionalidades, antigas e novas, com

isso, os *stakeholders* têm uma visão objetiva do andamento e desenvolvimento do programa e as equipes recebem um rápido *feedback*;

- *Inspect and Adapt* (I&A): Ao final de cada PI, o atual estado da solução é demonstrado e avaliado. As equipes refletem e identificam os itens do *backlog* que podem ser melhorados através de *workshops*, que são ótimas oportunidades para coletar dados, resolver problemas e buscar ações que possam aumentar a velocidade, a qualidade e a confiabilidade para o próximo PI, possibilitando que os ARTs evoluam constantemente. Sempre que possível, todos os "passageiros do trem" (equipes, RTE, *Business Owners* e entre outros papéis) devem participar das etapas desses *workshops*:
 - Apresentar Demo do PI (I&A01): é diferente das demais demonstrações, uma vez que limita-se a mostrar apenas as funcionalidades desenvolvidas durante o PI. Durante a apresentação, os *business owners*, clientes e outros interessados ajudam cada equipe a avaliar o valor de negócio alcançado;
 - Revisão dos indicadores quantitativos (I&A02): as equipes coletam e discutem sobre os dados e as tendências observadas. Uma medida discutida é a Medida de Previsibilidade do Programa, que é a avaliação dos reais valores de negócio atingidos pelos objetivos do PI de cada equipe, análise feita conjuntamente por membros das equipes, *business owners*, clientes e seus representantes, além de outros interessados principais, durante a apresentação da demo. Uma boa taxa de realização, acima de 80%, indica boa confiabilidade para o trem;
 - Retrospectiva (I&A03): as equipes realizam uma breve retrospectiva sobre o PI, buscando identificar questões e problemas que queiram abordar. Com os temas levantados, os participantes elegem os que serão tratados, geralmente, problemas maiores do nível de Programa são escolhidos ao invés dos de nível de Equipe. Com isto decidido, inicia-se o *Workshop* de resolução de problema(s);
 - Concordar sobre o problema abordado (I&A04): o objetivo desta atividade é alinhar a perspectiva de todos sobre o problema. As visões sobre do que se trata, onde ocorre e impactos que causam devem ser expostas sucintamente;
 - Analisar a Raiz do Problema (I&A05): recomenda-se o uso de um diagrama de Ishikawa, também conhecido como espinha de peixe, que propõe o levantamento de causas e sub-causas das categorias principais que no caso são pessoas, processos, ferramentas, programa e ambiente. A equipe então procura levantar as causas que contribuíram para que tal problema ocorresse. Uma vez identificadas, procura-se reconhecer suas raízes utilizando a técnica dos 5 porquês, técnica que consiste em perguntar 5 vezes o motivo do acontecimento do problema. Repete-se o método em cada raiz encontrada pelo menos cinco vezes, permitindo que as raízes mais profundas dos problemas sejam identificadas[18];

- Identificar a Maior Causa Raiz (I&A06): os membros das equipes votam na causa que acham ser a maior raiz do problema. Um gráfico é feito apresentando os votos recebidos por cada causa;
 - Realizar *Brainstorm* (I&A07): são levantadas o máximo de ações corretivas para solucionar a raiz do problema;
 - Criar Itens de Melhoria para o *Backlog* (I&A08): as equipes votam nas três soluções mais prováveis, que serão transformadas em histórias e recursos para a próxima reunião de planejamento de PI, onde o RTE garante que essas melhorias serão repassadas para os *backlogs* dos times;
- Gerenciar Fluxo do ART (GFA01): responsável por controlar o fluxo contínuo de entrega, integração e implantação de incrementos, além de identificar gargalos e oportunidades para melhorias. Sugere-se a utilização do Kanban para visualizar o progresso das atividades, além de ser uma maneira fácil de mostrá-lo aos *stakeholders*. Um exemplo típico de Kanban pode ser visto na Figura 3.

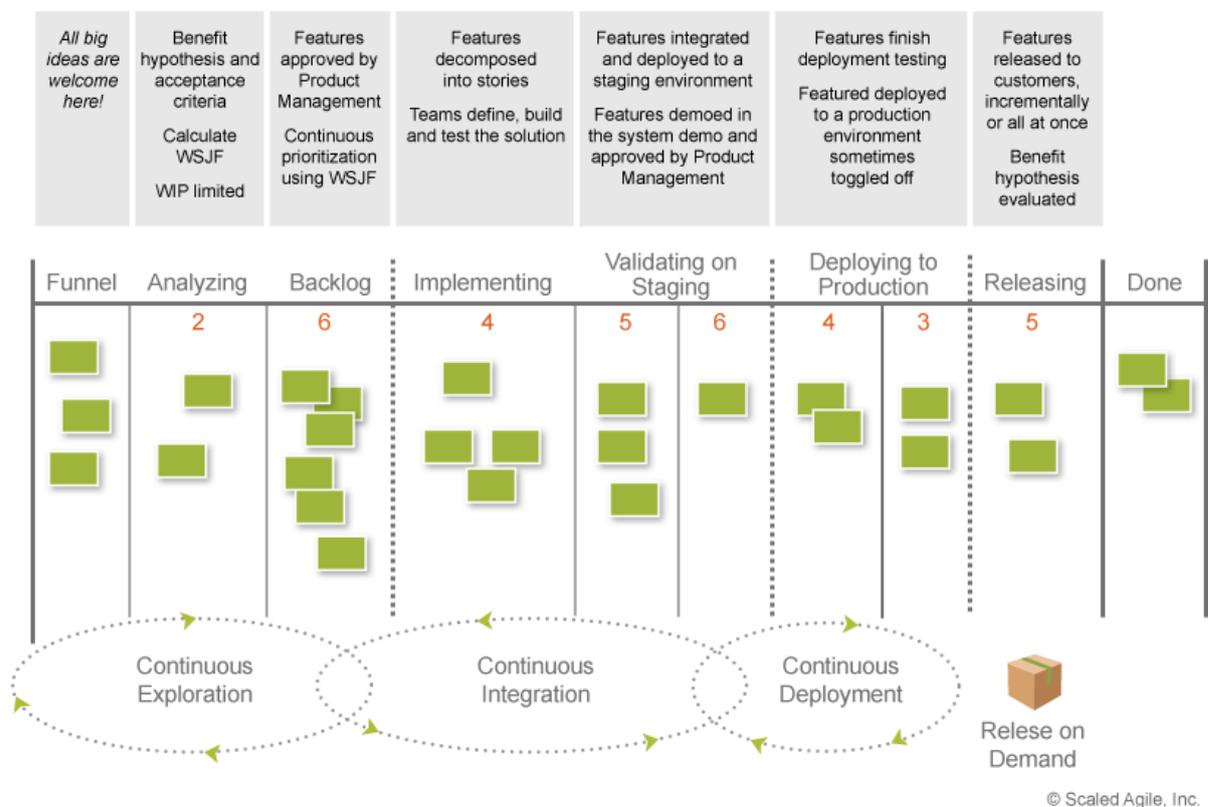


Figura 3 – Exemplo de utilização do Kanban no nível de programa. (Fonte: [2])

3.3 *Portfolio Level*

O nível de Portfólio contem os princípios, práticas e papéis necessários para iniciar e governar um conjunto de fluxos de valor. A estratégia e os investimentos financeiros são definidos nesse nível, utilizando práticas propostas pelo *Lean Thinking* alinhadas com práticas ágeis.

Os fluxos de valor (*Value Streams*), representam a série de etapas que uma organização segue para criar soluções, fornecendo um fluxo contínuo de valor para um cliente. Os fluxos de valor são utilizados para definir e atingir os objetivos comerciais do Portfólio, e organizar os ARTs para oferecer funcionalidades mais rapidamente.

Empresas maiores podem possuir vários portfólios, um para cada instância do SAFe, sendo que cada portfólio possui uma conexão de duas vias com a instituição, estabelecendo os temas estratégicos para o portfólio, que são objetivos de negócio específicos e detalhados que conectam o portfólio aos objetivos estratégicos da empresa, e oferecendo um *feedback* constante sobre o portfólio aos *stakeholders*. As informações presentes no *feedback* incluem o estado atual das soluções do portfólio, indicadores do desempenho dos fluxos de valores (KPIs), avaliações qualitativas da solução atual para o mercado, qualquer força, fraqueza, ameaça ou oportunidade encontrada no nível.

Os papéis no Portfólio incluem cargos de grande responsabilidade e governança:

- *Lean Portfolio Management*: Os indivíduos com esse papel possuem o mais alto nível de tomada de decisão e responsabilidade financeira em um portfólio SAFe. São responsáveis por três áreas principais: estratégia e financiamento de investimentos, orientação ágil e governança Lean;
- *Epic Owners*: São responsáveis por coordenar os *epics*, soluções grandes que exigem uma análise e aprovação financeira antes de serem transformadas em *features*, divididas em histórias de usuário e implementadas. O escopo de um *epic* deve envolver uma área da instituição, a finalidade da solução e a importância dessa solução. Dessa forma, um exemplo de escopo seria: O departamento de vendas quer integrar seus canais de venda para melhorar o atendimento ao cliente.
- *Enterprise Architect*: Responsável por coordenar múltiplos fluxos de valor, ajudando na estratégia para otimizar os resultados do Portfólio.

Com a perspectiva de que cada portfólio existe com o propósito de ser um conjunto de soluções alinhadas com a estratégia do negócio, entende-se que devem trabalhar dentro de um orçamento aprovado, já que os custos operacionais no desenvolvimento das soluções é um fator importante para o sucesso financeiro da empresa. Assim, nessa etapa, é adotada a prática de *Lean Budgets*, que minimiza os custos indiretos através do financiamento

dos fluxos de valor e não dos projetos, permitindo uma tomada de decisão rápida pelos responsáveis que acompanham o dia-a-dia do desenvolvimento.

Alguns problemas que levaram à escolha pelo *Lean Budget*, ao invés do método tradicional de financiamento de projetos, foram: a lentidão e complexidade em elaborar o orçamento, pois envolvem vários departamentos da empresa que necessitam ser analisados individualmente; as equipes terem que tomar decisões sobre prazos e quantidade de pessoas sem terem ainda conhecimento profundo sobre o projeto e as tarefas envolvidas; e a falta de flexibilidade e adaptação às mudanças de prioridade, o que leva a um re-orçamento, ocorrendo o mesmo com os atrasos no desenvolvimento de tarefas.

Dessa forma, o *Lean Budgets* propõe maior rapidez, adaptabilidade e transparência na distribuição de investimentos. Esses objetivos são alcançados pois não há a necessidade de re-orçar projetos inteiros; empodera-se papéis que acompanham o dia-a-dia do desenvolvimento, como o *Product Management*, para que decisões sejam tomadas rapidamente e para verificar se o investimento está sendo bem aplicado; e oferece-se a aos financiadores a possibilidade de acompanharem o desenvolvimento das soluções através de demonstrações feitas ao final de cada PI. Os responsáveis por gerenciar todo esse processo fazem parte do *Lean Portfolio Management*, definindo os orçamentos de cada *Value Stream* e de cada PI que o compõe.

As atividades realizadas nesta etapa são:

- Realizar Reunião Estratégica (PTF01): realizada entre os executivos da empresa e os responsáveis pelo portfólio, tem como objetivo aprovar orçamentos e definir os temas estratégicos. Para isso, analisam o orçamento operacional total, os objetivos de desempenho financeiro como participação no mercado e rentabilidade, as missões e valores da empresa, além da análise da competitividade do mercado;
- Realizar o Orçamento (PTF02): cada *Value Stream* pertencente ao projeto recebe um orçamento para cada PI, seguindo o método *Lean Budgets*;
- Realizar Acompanhamento (PTF03): utiliza-se um sistema de Kanban para visualizar, gerenciar e analisar o andamento do projeto, desde a idealização até a conclusão. Uma parte importante deste processo são os *Epics*, que por serem grandes e poderem afetar mais de um ART, necessitam de mais atenção. Assim, o Kanban do nível de portfólio é desenvolvido para idealizar, analisar, aprovar e rastrear os *Epics*, principalmente.

4 MODELO DE REFERÊNCIA MPS-BR DE SOFTWARE

Como exposto no capítulo 2, utiliza-se modelos de referência para mensurar a maturidade dos processos de desenvolvimento e assim, obter auxílio para implementar melhorias e aumentar a qualidade do processo como um todo.

Um dos modelos mais utilizados mundialmente é o CMMI (*Capability Maturity Model Integration*), desenvolvido pelo *Software Engineering Institute* nos Estados Unidos[8]. O modelo abrange todo o ciclo de produção, desde a concepção até a entrega e manutenção do *software*, apresentando o caminho evolutivo para que a organização alcance um processo maduro.

Apesar de todo reconhecimento, há discussões sobre sua aplicabilidade pelas pequenas empresas[19]. A dificuldade de sua aplicação também é encontrada nas organizações brasileiras, devido a complexidade do modelo e aos altos custo para obtenção da certificação.

Visando a adaptação à realidade das organizações brasileiras, foi desenvolvido, pela Associação para Promoção da Excelência do *Software* Brasileiro (SOFTEX), o MPS.BR. O modelo nacional tem como objetivo melhorar, a longo prazo, a capacidade de desenvolvimento de *software*, serviços e práticas de gestão de RH das indústrias de TIC.[5]

Algumas vantagens da utilização do MPS.BR, segundo [20], são: os sete níveis do modelo permitem uma implementação mais gradual, sendo adequado às micros, pequenas e médias empresas de *software*; aplicação a custos acessíveis e adequados à realidade brasileira; a qualificação da empresa que utiliza o modelo é mantida por avaliações periódicas realizadas de dois em dois anos.

O MPS.BR possui cinco componentes, dentre eles o modelo de referência MPS para *Software* (MR-MPS-SW). Esse modelo tem como base técnica a NBR ISO/IEC 12207[9], o CMMI-DEV[®][21] e a norma internacional ISO/IEC 33020[22].

O MR-MPS-SW define sete níveis de maturidade, que são uma combinação entre processos e sua capacidade, variando de A a G: A (Em otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). A escala de maturidade se inicia em G e progride até o nível A.

A capacidade de um processo é a caracterização de sua habilidade em atender os atributos de processo (APs) associados aos processos de cada nível de maturidade, expressando o grau de refinamento e institucionalização com que o processo está sendo executado na organização. Os níveis são acumulativos, ou seja, se a organização estiver no

nível F, ela deve atender aos APs do nível G e F[5]. Os APs presentes no MR-MPS-SW são:

- AP 1.1 – O processo é executado;
- AP 2.1 – A execução do processo é gerenciada;
- AP 2.2 – Os produtos de trabalho do processo são gerenciados;
- AP 3.1 – O processo é definido;
- AP 3.2 – O processo está implementado;
- AP 4.1 – O processo é objeto de análise quantitativa;
- AP 4.2 – O processo é controlado quantitativamente;
- AP 5.1 – O processo é objeto de melhorias incrementais e inovações;
- AP 5.2 – O processo é objeto de implementação de melhorias inovadoras e incrementais.

A tabela 2 apresenta a relação entre níveis de maturidade do MR-MPS-SW, processos e atributos de processos.

Como o escopo deste trabalho não envolve a execução dos processos em uma organização, os atributos do processo não serão detalhados e nem utilizados na análise e alinhamento do processo de desenvolvimento da GAIA.

Para melhor entendimento da Tabela 2, serão apresentados os aspectos e objetivos de cada processo na próxima subseção. No MR-MPS-SW, os processos são analisados em termos de resultados esperados, sendo que cada processo possui um conjunto diferente. Como o escopo deste trabalho envolve apenas os níveis G e F, apenas seus resultados esperados serão expostos[17].

4.0.1 Descrição dos processos

Os processos presentes no MR-MPS-SW são[5]:

- Gerência de Projetos – GPR: estabelece e mantém planos que definem as atividades, recursos e responsabilidades do projeto. Também provê informações sobre o andamento do projeto, permitindo que correções sejam feitas quando desvios significativos no desempenho do projeto são identificados;
- Gerência de Requisitos – GRE: gerencia os requisitos do produto e de seus componentes ao longo do projeto, identificando inconsistências e ajustando as mudanças para manter uma boa relação bidirecional entre os requisitos e os produtos;

- Aquisição – AQU: gerencia a aquisição de produtos que satisfaçam às expectativas do comprador;
- Gerência de Configuração – GCO: define e acompanha as configurações dos produtos envolvidos no projeto, identificando, modificando, armazenando e disponibilizando a todos os envolvidos;
- Garantia de Qualidade – GQA: assegura que os produtos, bem como a execução dos processos, estejam conforme o planejado e seguindo os procedimentos e padrões estabelecidos;
- Gerência de Portfólio de Projetos – GPP: inicia e mantém projetos necessários, suficientes e sustentáveis para alcançar os objetivos estratégicos da organização. Também é responsável pela distribuição de investimentos, realizando uma qualificação contínua de projetos para verificar se justificam os investimentos recebidos;
- Medição – MED: coleta, armazena e analisa os dados dos produtos, processos e projetos da organização;
- Avaliação e Melhoria do Processo Organizacional – AMP: determina o quanto os processos padrão da organização contribuem para seu sucesso, apoiando a organização a melhorá-los;
- Definição do Processo Organizacional – DFP: estabelece e mantém um conjunto de ativos de processo organizacional e padrões que atendam as necessidades da organização;
- Gerência de Recursos Humanos – GRH: provê e mantém adequados os recursos humanos necessários à organização e aos projetos;
- Gerência de Reutilização – GRU: gerencia o ciclo de vida dos ativos reutilizáveis;
- Desenvolvimento de Requisitos – DRE: define os requisitos do cliente, do produto e de seus componentes;
- Integração do Produto – ITP: compõe os componentes do produto a fim de produzir um produto integrado com seu projeto e demonstrar que os requisitos foram satisfeitos para o ambiente alvo;
- Projeto e Construção do Produto – PCP: projeta, desenvolve e implementa soluções para atender aos requisitos;
- Validação – VAL: confirma que um produto ou um componente de produto atenderá à seu uso pretendido quando colocado no ambiente para o qual foi desenvolvido;

- Verificação – VER: confirma que cada serviço e/ou produto do processo atendem ao requisitos especificados;
- Desenvolvimento para Reutilização – DRU: identifica oportunidades de reutilização dos ativos na organização. Se possível, também desenvolvem ativos que possam ser utilizados futuramente;
- Gerência de Decisões – GDE: analisa decisões críticas utilizando um processo formal para avaliação das alternativas identificadas;
- Gerência de Riscos – GRI: identifica, analisa, trata, monitora e procura reduzir continuamente os riscos de projetos, bem como da organização como um todo.

Tabela 2 – Níveis de maturidade do MR-MPS-SW

Nível	Processos	Atributos de processos
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, AP 4.2
C	Gerência de Riscos - GRI Desenvolvimento para Reutilização - DRU Gerência de Decisões	AP 1.1, AP 2.1, AP 2.2, AP 3.1, 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, 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	AP1.1, AP 2.1, AP 2.2, AP 3.1, 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, AP 2.2
G	Gerência de Requisitos - GRE Gerência de Projetos - GPR	AP 1.1, AP 2.1

Fonte: [SOFTEX\[5\]](#)

4.0.2 Resultados esperados no nível G

Como apresentado na Tabela 2, o nível G envolve os processos de Gerência de Projetos e Gerência de Requisitos. Os resultados esperados para cada processo são apresentados nas Tabelas 3 e 4, respectivamente.

Tabela 3 – Resultados esperados para o processo de Gerência de Projetos

GPR01	O escopo do trabalho para o projeto é definido.
GPR02	As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados.
GPR03	O modelo e as fases do ciclo de vida do projeto são definidos.
GPR04	O esforço e o custo para a execução das tarefas e dos produtos são estimados com base em dados históricos ou referências técnicas.
GPR05	O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos.
GPR06	Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados.
GPR07	Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo.
GPR08	Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados.
GPR09	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.
GPR10	Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos.
GPR11	A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados.
GPR12	O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido.
GPR13	O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado.
GPR14	Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado.
GPR15	Os riscos são monitorados em relação ao planejado.
GPR16	O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido.
GPR17	Revisões são realizadas em marcos do projeto conforme estabelecido no planejamento.
GPR18	Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas.
GPR19	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.

Fonte: [SOFTEX\[5\]](#)

Tabela 4 – Resultados esperados para o processo de Gerência de Requisitos

GRE01	O entendimento dos requisitos é obtido junto aos fornecedores de requisitos.
GRE02	Os requisitos são avaliados com base em critérios objetivos e um comprometimento da equipe técnica com estes requisitos é obtido.
GRE03	A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida.
GRE04	Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos.
GRE05	Mudanças nos requisitos são gerenciadas ao longo do projeto.

Fonte: [SOFTEX\[5\]](#)

4.0.3 Resultados esperados no nível F

Os processos envolvidos no nível F são Aquisição, Gerência de Configurações, Garantia de Qualidade, Gerência de Portfólio de Projetos e Medição. Seus resultados esperados são descritos nas Tabelas de 5 a 9.

Tabela 5 – Resultados esperados para o processo de Aquisição

AQU01	As necessidades de aquisição, as metas, os critérios de aceitação do produto, os tipos e a estratégia de aquisição são definidos.
AQU02	Os critérios de seleção do fornecedor são estabelecidos e usados para avaliar os potenciais fornecedores.
AQU03	O fornecedor é selecionado com base na avaliação das propostas e dos critérios estabelecidos.
AQU04	Um acordo que expresse claramente as expectativas, responsabilidades e obrigações de ambas as partes (cliente e fornecedor) é estabelecido e negociado entre elas.
AQU05	Um produto que satisfaça a necessidade expressa pelo cliente é adquirido baseado na análise dos potenciais candidatos.
AQU06	A aquisição é monitorada de forma que as condições especificadas sejam atendidas, tais como custo, cronograma e qualidade, gerando ações corretivas quando necessário.
AQU07	O produto é entregue e avaliado em relação ao acordado e os resultados são documentados.
AQU08	O produto adquirido é incorporado ao projeto, caso pertinente.

Fonte: [SOFTEX\[5\]](#)

Tabela 6 – Resultados esperados para o processo de Gerência de Configuração

GCO01	Um Sistema de Gerência de Configuração é estabelecido e mantido.
GCO02	Os itens de configuração são identificados com base em critérios estabelecidos.
GCO03	Os itens de configuração sujeitos a um controle formal são colocados sob baseline.
GCO04	A situação dos itens de configuração e das baselines é registrada ao longo do tempo e disponibilizada.
GCO05	Modificações em itens de configuração são controladas.
GCO06	O armazenamento, o manuseio e a liberação de itens de configuração e baselines são controlados.
GCO07	Auditorias de configuração são realizadas objetivamente para assegurar que as baselines e os itens de configuração estejam íntegros, completos e consistentes.

Fonte: [SOFTEX\[5\]](#)

Tabela 7 – Resultados esperados para o processo de Garantia de Qualidade

GQA01	A aderência dos produtos de trabalho aos padrões, procedimentos e requisitos aplicáveis é avaliada objetivamente, antes dos produtos serem entregues e em marcos predefinidos ao longo do ciclo de vida do projeto.
GQA02	A aderência dos processos executados às descrições de processo, padrões e procedimentos é avaliada objetivamente.
GQA03	Os problemas e as não-conformidades são identificados, registrados e comunicados.
GQA04	Ações corretivas para as não-conformidades são estabelecidas e acompanhadas até as suas efetivas conclusões. Quando necessário, o escalamento das ações corretivas para níveis superiores é realizado, de forma a garantir sua solução.

Fonte: [SOFTEX\[5\]](#)

Tabela 8 – Resultados esperados para o processo de Gerência de Portfólio de Projetos

GPP01	As oportunidades de negócio, as necessidades e os investimentos são identificados, qualificados, priorizados e selecionados em relação aos objetivos estratégicos da organização por meio de critérios objetivos.
GPP02	Os recursos e orçamentos para cada projeto são identificados e alocados.
GPP03	A responsabilidade e autoridade pelo gerenciamento dos projetos são estabelecidas.
GPP04	O portfólio é monitorado em relação aos critérios que foram utilizados para a priorização.
GPP05	Ações para corrigir desvios no portfólio e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão.
GPP06	Os conflitos sobre recursos entre projetos são tratados e resolvidos, de acordo com os critérios utilizadas para a priorização.
GPP07	Projetos que atendem aos acordos e requisitos que levaram à sua aprovação são mantidos, e os que não atendem são redirecionados ou cancelados.
GPP08	A situação do portfólio de projetos é comunicada para as partes interessadas, com periodicidade definida ou quando o portfólio for alterado.

Fonte: [SOFTEX](#)[5]

Tabela 9 – Resultados esperados para o processo de Medição

MED01	Objetivos de medição são estabelecidos e mantidos a partir dos objetivos de negócio da organização e das necessidades de informação de processos técnicos e gerenciais.
MED02	Um conjunto adequado de medidas, orientado pelos objetivos de medição, é identificado e definido, priorizando, documentado, revisado e, quando pertinente, atualizado.
MED03	Os procedimentos para a coleta e o armazenamento de medidas são especificados.
MED04	Os procedimentos para a análise das medidas são especificados.
MED05	Os dados requeridos são coletados e analisados.
MED06	Os dados e os resultados das análises são armazenados.
MED07	Os dados e os resultados das análises são comunicados aos interessados e são utilizados para apoiar decisões.

Fonte: [SOFTEX](#)[5]

5 PROCESSO DE DESENVOLVIMENTO DE *SOFTWARE* GAIA

O Laboratório GAIA - Soluções em TIC do Departamento de Computação da UEL desenvolve trabalhos em temas como infraestrutura, processos, sistemas e recursos humanos, a fim de auxiliar o uso de TI nas organizações. Estão envolvidos nesses trabalhos desde consultorias até o desenvolvimento de soluções inovadoras, sempre baseando-se na experiência acumulada em pesquisas acadêmicas realizadas na área[23].

Nesse contexto, foi criado o processo de Desenvolvimento de *software* (PDS) GAIA[3] baseado em uma perspectiva de gerenciamento do PMBOK. Organizado pelo *Project Management Institute* (PMI), o PMBOK é um conjunto de práticas na gestão de projetos que conceitua e identifica processos, áreas de conhecimento, ferramentas e técnicas, sendo considerado por muitos profissionais da área como sendo a base para o conhecimento de gestão de projetos[24].

Assim, o PDS GAIA possui 6 fases: Análise Inicial, Análise e Planejamento, Execução e Implementação, Validação e Testes, Entrega e Finalização. Além dessas etapas, o PDS é composto por mais duas componentes que atuam paralelamente aos demais, o Manter Requisitos e Gerenciar Portifólio.

Como pode ser visto na Figura 4, o processo começa pela Análise Inicial, cujas tarefas são focadas em estabelecer o escopo e a viabilidade do projeto. Em seguida, ocorre a Análise e Planejamento que revisa e valida requisitos, define tarefas, elabora cronogramas e define prazos.

A próxima etapa é a Execução e Implementação, que envolve a gerência de riscos e a garantia da qualidade, além das tarefas referentes à execução do projeto. Após a implementação ocorre a Validação e Testes, que, como um ciclo, volta para a fase de execução caso ocorra falhas. Paralelamente à essas duas etapas há o Manter Requisitos, que recebe, analisa e define o impacto de alterações nos requisitos e atualiza os documentos necessários.

Após a validação e os testes serem concluídos, ocorre a Entrega, que realiza testes de integração, analisa resultados, executa correções e realiza reuniões de *feedback* com os clientes. Por último, acontece a Finalização, onde é feita a entrega final e uma confraternização.

O Gerenciar Portifólio é responsável por manter o portfólio de produtos e serviços da organização, realizando reuniões estratégicas e de acompanhamento.

A fim de estudar o alinhamento do PDS GAIA ao *framework* ágil SAFe e, posteriormente, ao modelo de qualidade MR-MPS-SW, serão detalhadas e rotuladas as tarefas

envolvidas nas etapas do PDS[17].

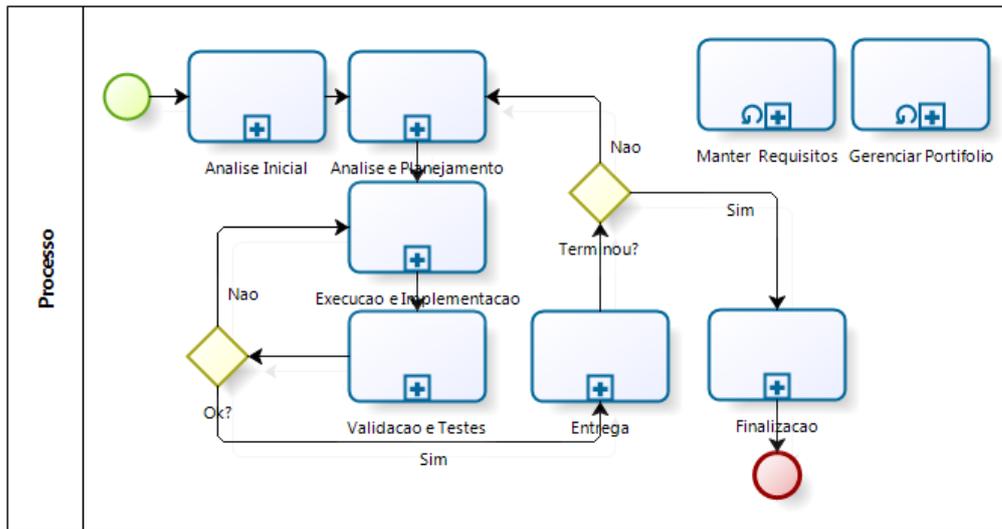


Figura 4 – Processo de Desenvolvimento de *Software* GAIA. (Fonte: [3])

5.1 Análise Inicial - ANI

A análise inicial reúne as tarefas necessárias para iniciar o projeto, procurando principalmente, definir o escopo do projeto e obter o comprometimento de todos os envolvidos. Seu fluxograma pode ser visto na Figura 5.

As tarefas envolvidas nessa etapa são:

- Reuniões - PDS-ANI01: são reuniões realizadas entre os responsáveis pelo fornecimento dos requisitos e os responsáveis por recebê-los e gerenciá-los. É guiada pela definição das datas e pautas das reuniões e criação das atas assinadas das mesmas;
- Estabelecer Escopo - PDS-ANI02: Um escopo inicial do projeto é desenvolvido, afim de guiar as próximas etapas. É feito através do levantamento das necessidades do cliente, onde se estabelece o escopo e gera-se um WBS simplificado do projeto;
- Realizar Estimativas - PDS-ANI03: é estimado o tamanho do projeto, através da identificação de suas características e seguindo técnicas específicas;
- Preparar ambiente - PDS-ANI04: prepara o ambiente para o desenvolvimento do projeto, incluindo seu planejamento, controle de acesso, versionamento, entre outros, de acordo com o padrão organizacional;
- Analisar Viabilidade - PDS-ANI05: Verifica-se a viabilidade do projeto, analisando riscos, premissas e restrições identificados durante a elaboração do escopo. São determinados também os prazos e custos para o projeto;

- Criar *Project Charter* - PDS-ANI06: se o projeto for considerado viável, reúne-se todas as informações disponíveis em um único documento (*Project Charter*);
- Realizar Reunião de *Kick-off* - PDS-ANI07: todos os participantes do projeto são reunidos a fim de se obter o comprometimento de todos. É definida a pauta da reunião e apresenta-se o *Project Charter*.

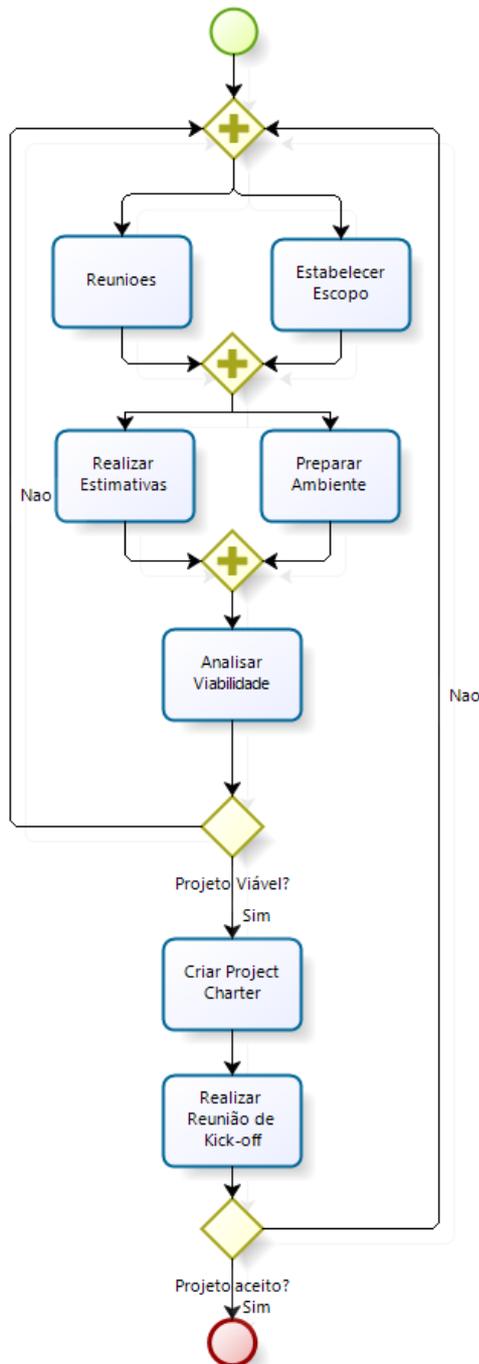


Figura 5 – Análise Inicial - ANI. (Fonte: [3])

5.2 Análise e Planejamento - APL

Essa etapa reúne as tarefas de análise e planejamento do projeto e de suas fases, sendo realizadas estimativas e planos. Seu fluxograma é extenso, por isso é apresentado dividido nas Figuras 6 e 7

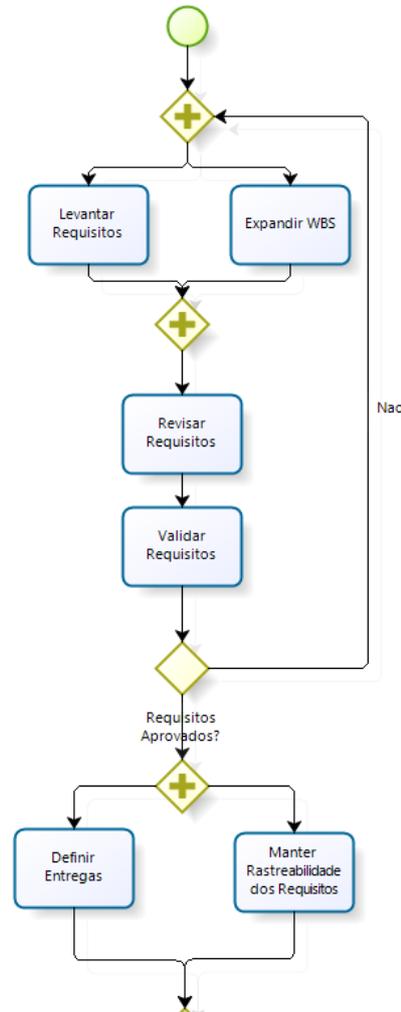


Figura 6 – Análise e Planejamento - APL(Parte 1). (Fonte: [3])

As tarefas envolvidas na primeira parte, Figura 6, desta etapa são:

- Levantar Requisitos - PDS-APL01: Necessidades e requisitos são levantados com o cliente. Procurando entender a situação, é realizada uma reunião entre os responsáveis pelos requisitos e o cliente, em seguida, os requisitos são elencados seguindo as necessidades e detalhados conforme o padrão da organização. Aqui também são modelados os casos de uso;
- Expandir WBS - PDS-APL02: o WBS é melhor detalhado. Atividades e tarefas são derivadas da original e o domínio da aplicação é esmiuçado;

- Revisar Requisitos - PDS-APL03: feito o levantamento e a expansão do WBS, os requisitos são reunidos, revisados e verificados seguindo critérios estabelecidos;
- Validar Requisitos - PDS-APL04: Os requisitos classificados e verificados são apresentados ao cliente, a fim de obter sua validação e questionamentos por quaisquer alterações;
- Definir Entregas - PDS-APL05: com os requisitos definidos, revisados e validados, o sistema especificado é dividido em entregas implementáveis para que seu desenvolvimento possa ocorrer de forma mais fluída e visível. É definido quais serão as entregas;
- Manter Rastreabilidade dos Requisitos - PDS-APL06: é criado um relacionamento bidirecional entre os requisitos. Relacionamentos bidirecionais entre requisitos e outros ativos de projeto também são criados.

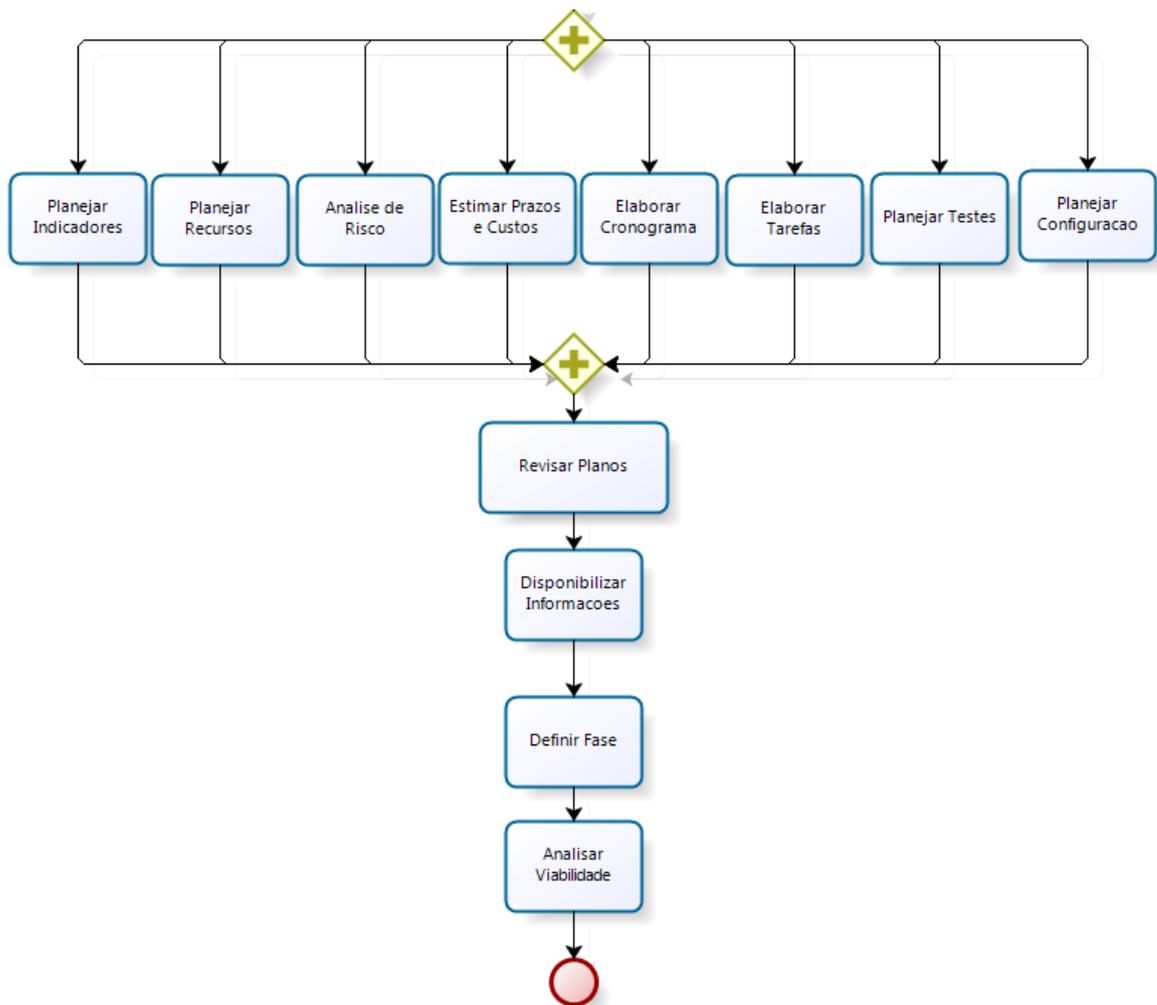


Figura 7 – Análise e Planejamento - APL(Parte 2). (Fonte: [3])

Na segunda parte da etapa, Figuras 7, estão as tarefas:

- Planejar Indicadores - PDS-APL07: indicadores de desempenho são identificados e estabelecidos. Para isso, são identificados, primeiramente, os objetivos de medição, obtidos com a ajuda da elaboração de questões que devem ser respondidas a partir dos dados coletados. Com os objetivos definidos, é definido a política de quais, quando e como os dados serão coletados, determinando prioridades, descrevendo métodos de coleta e criando um Plano de Métricas;
- Planejar Recursos - PDS-APL08: os recursos necessários para a realização das tarefas são identificados, sua disponibilidade é verificada e seu orçamento realizado. Por fim, uma lista com os recursos é criada e um plano de alocação de recursos é feito;
- Análise de Risco - PDS-APL09: os riscos potenciais do projeto são identificados e qualificados de acordo com sua probabilidade de ocorrência. Os riscos também são qualificados segundo o impacto que podem causar no projeto. São elaboradas ações preventivas e corretivas para os riscos identificados, além de planos de contingência. Todas essas ações são organizadas então em uma única lista;
- Estimar Prazos e Custos - PDS-APL10: tarefas e recursos são elencados e dados de custos indiretos e de RH são coletados. Com essas informações, é possível estimar a duração das atividades e dos recursos, permitindo a realização de uma previsão de custo efetivo do projeto no geral;
- Elaborar Cronograma - PDS-APL11: atividades, recursos, estimativas de prazos e custos são reunidos, adicionando uma margem para erros. Pontos de controle são definidos e então, o cronograma é elaborado;
- Elaborar Tarefas - PDS-APL12: atividades e tarefas são identificadas e organizadas em uma lista;
- Planejar Testes - PDS-APL13: testes necessários para garantir a qualidade dos produtos parciais são pensados e então, é determinado o que testar, como os testes serão executados e como analisá-los;
- Planejar Configuração - PDS-APL14: define o plano de gerenciamento de configuração. São definidos os ativos que serão gerenciados, o local de armazenamento, a configuração inicial de arquivos, pontos de revisão, ferramentas de gerenciamento de configuração e pontos de criação das *baselines*;
- Revisar Planos - PDS-APL15: os planos criados até agora são revisados em sua consistência. A coerência de cada plano é verificada separadamente, para depois ser verificada como uma unidade. Os objetivos e aderência dos planos são analisados e, por fim, relacionamentos entre os mesmos são criados;

- Disponibilizar Informações - PDS-APL16: as informações são reunidas na ferramenta de configuração, disponibilizadas, em um único ambiente, a todos interessados, e, opcionalmente, restrições de acesso podem ser aplicadas;
- Definir Fase - PDS-APL17: A fase a ser desenvolvida é definida através da seleção das entregas implementáveis a serem realizadas;
- Analisar Viabilidade - PDS-APL18: identifica-se o que falta ser realizado e seus riscos para, então, analisar a viabilidade do projeto.

5.3 Execução e Implementação - EXI

Nessa etapa estão envolvidas tarefas referentes à execução do projeto, controlando suas tarefas e andamento geral. Seu fluxograma é apresentado na Figura 8

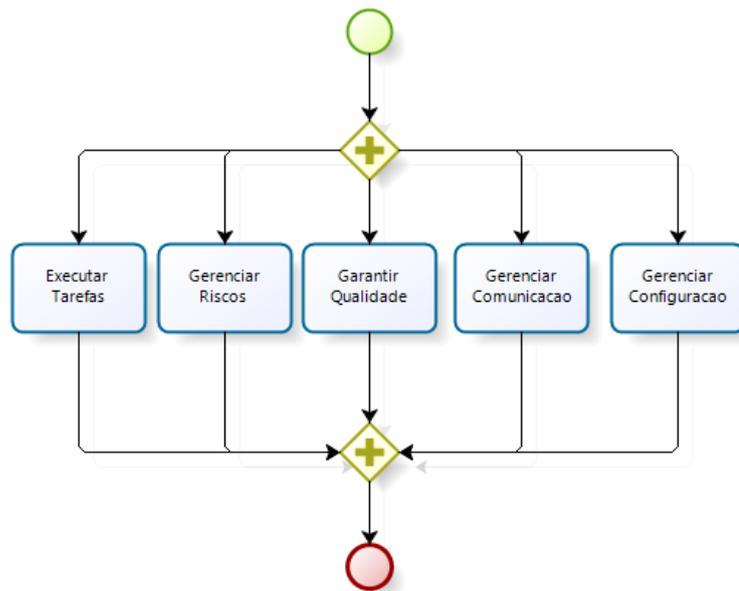


Figura 8 – Execução e Implementação - EXI. (Fonte: [3])

As tarefas envolvidas nessa etapa são:

- Executar Tarefas - PDS-EXI01: tarefas planejadas anteriormente são executadas e atualizadas em seu marcador de progresso;
- Gerenciar Riscos - PDS-EXI02: O desenvolvimento do projeto é monitorado e seus riscos controlados através da identificação dos riscos potenciais, de acordo com o plano de riscos, execução de ações preventivas e corretivas, e registro dos riscos ocorridos;

- Garantir Qualidade - PDS-EXI03: o código é verificado, padrões são controlados, o *software* versionado, são implementados planos de testes, dados do projeto e processo são coletados e analisados, tudo para garantir que os produtos estão em conformidade com o esperado e com os padrões, além de garantir que o processo está sendo seguido;
- Gerenciar Comunicação - PDS-EXI04: garante que as informações necessárias são entregues às devidas pessoas no tempo certo;
- Gerenciar Configuração - PDS-EXI05: revisa os *baselines* e os ativos de projeto que estão sob a gerência de configuração.

5.4 Validação e Testes - VAT

A etapa de Validação e Testes é responsável por reunir as atividades referentes ao trabalho de testar unitariamente os resultados da fase executada. A Figura 9 apresenta o fluxograma da etapa.

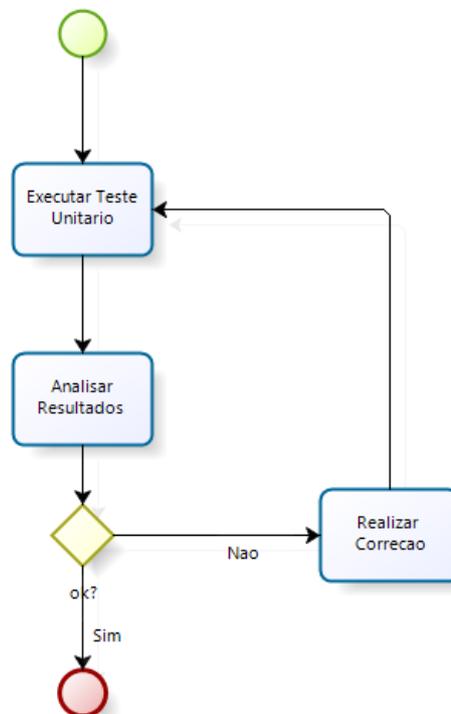


Figura 9 – Validação e Testes - VAT. (Fonte: [3])

As tarefas envolvidas nessa etapa são:

- Executar Teste Unitário - PDS-VAT01: testes unitários são realizados a fim de evitar erros de código e validar a fase;

- Analisar Resultados - PDS-VAT02: Analisa os resultados dos testes e valida o produto;
- Realizar Correção - PDS-VAT03: efetua correções necessárias de acordo com a análise dos resultados dos testes.

5.5 Entrega - ENT

A etapa reúne as tarefas necessárias para implantar o sistema no cliente e obter seu *feedback*. Seu fluxograma pode ser visto na Figura 10.

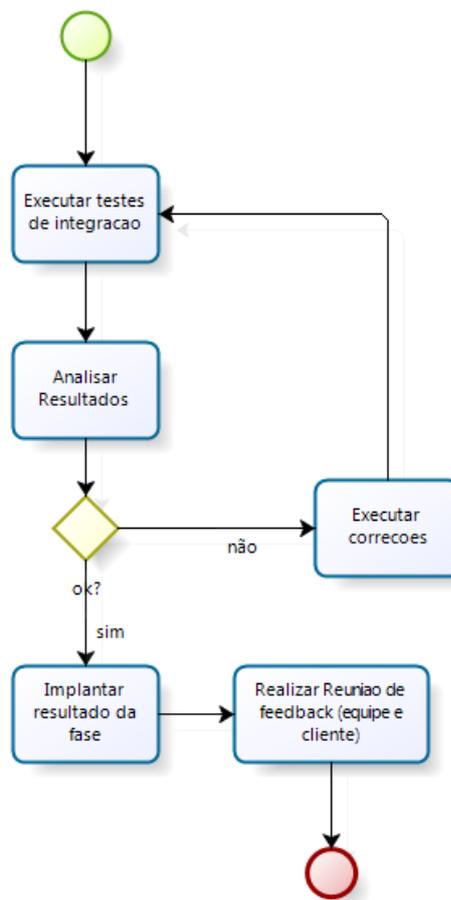


Figura 10 – Entrega - ENT. (Fonte: [3])

As tarefas envolvidas nessa etapa são:

- Executar Testes de Integração - PDS-ENT01: testes de integração são realizados, segundo o plano de testes, para evitar erros no sistema;
- Analisar Resultados - PDS-ENT02: analisa os resultados do teste de integração, identifica erros e valida o produto;

- Executar Correções - PDS-ENT03: correções são feitas conforme os resultados da análise;
- Implantar Resultado da Fase - PDS-ENT04: com a aprovação nos testes, é possível realizar a implantação do resultado da fase no cliente, reunindo os recursos necessários e configurando e testando seu funcionamento;
- Realizar Reunião de Feedback (equipe e cliente) - PDS-ENT05: reunião para formalizar a entrega do resultado da fase para o cliente e receber *feedback*. Pode ocorrer a aprovação ou não do cliente.

5.6 Finalização - FIN

Atividades necessárias para finalização do projeto, incluindo o contrato, são reunidas nesta etapa. Seu fluxograma pode ser visto na Figura 11.



Figura 11 – Finalização - FIN. (Fonte: [3])

As tarefas envolvidas nessa etapa são:

- Realizar Reunião de Entrega - PDS-FIN01: formaliza a entrega do produto ao cliente e o *feedback* do produto. Reúne-se informações e as apresenta em reunião, as lições aprendidas são capturadas e o *feedback* obtido e registrado;
- Confraternização - PDS-FIN02: realiza-se uma confraternização para comemorar o fim do projeto.

5.7 Manter Requisitos - MRQ

Nessa etapa reúne-se as atividades ligadas diretamente ao gerenciamento de requisitos. Seu fluxograma pode ser visto na Figura 12.

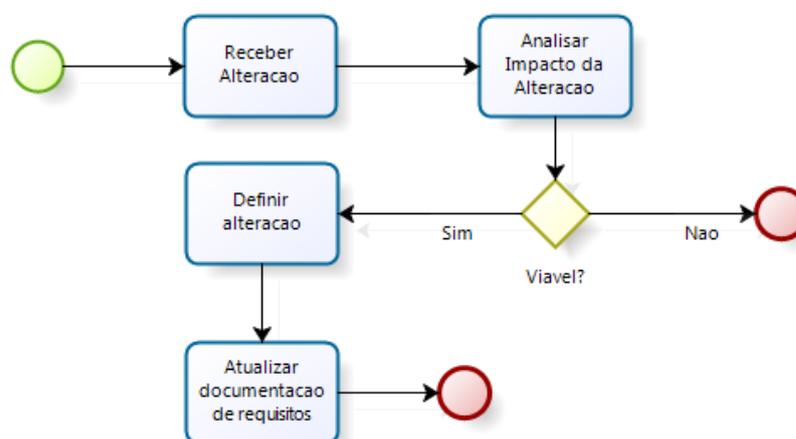


Figura 12 – Manter requisitos - MRQ. (Fonte: [3])

As tarefas envolvidas nessa etapa são:

- Receber Alteração - PDS-MRQ01: o cliente solicita alterações de requisitos. As alterações podem ser recebidas pelo *help-desk*, são registradas e compreendidas. Uma reunião pode ser marcada com o cliente;
- Analisar Impacto da Alteração - PDS-MRQ02: as alterações recebidas são avaliadas junto à equipe. Discute-se sobre a viabilidade das alterações;
- Definir Alteração - PDS-MRQ03: se a alteração for viável, é definido como e quando ela será feita;
- Atualizar Documentação de Requisitos - PDS-MRQ04: Os produtos de trabalho são atualizados de acordo com as alterações definidas.

5.8 Gerenciar Portfólio - GPT

As atividades necessárias para gerenciar o portfólio de produtos e serviços da organização são reunidas nessa etapa. A Figura 13 traz o fluxograma da etapa.

As tarefas envolvidas nessa etapa são:

- Realizar Reunião Estratégica - PDS-GPT01: essas reuniões são realizadas com o intuito de tomar decisões estratégicas, identificando e priorizando oportunidades.

Ocorre também o direcionamento de investimentos e responsabilidades no gerenciamento de projetos, a atribuição das responsabilidades aos gerentes de projeto e a priorização dos investimentos;

- Realizar Reunião de Acompanhamento - PDS-GPT02: essas reuniões são realizadas a fim de obter informações sobre o andamento dos projetos atuais e identificar possíveis ações necessárias. A partir dessas informações, é possível identificar e planejar soluções.

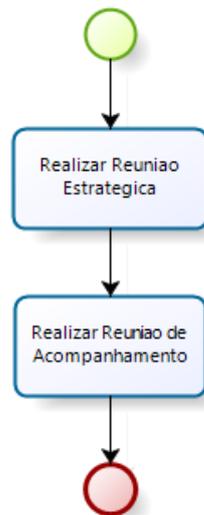


Figura 13 – Gerenciar Portfólio-GPT. (Fonte: [3])

6 ALINHAMENTO DO PDS GAIA

6.1 Metodologia

Para realizar o processo de alinhamento, primeiramente foram analisadas as características e detalhes da metodologia ágil SAFe (capítulo 3), dos processos do modelo de referência para melhoria do processo de *software* MR-MPS-SW (capítulo 4) e de seus resultados esperados para os níveis G e F, e das atividades realizadas pelo PDS da GAIA (capítulo 5).

Em seguida, foram procuradas evidências, nas atividades rotuladas do PDS GAIA, da realização dos processos propostos pelo SAFe e dos resultados esperados do MR-MPS-SW. Através das evidências, foi possível classificar os processos levando em consideração o quão implementados estão no PDS. Essa classificação seguiu o Grau de Rigoriedade proposto na subseção 6.1.1. Os processos e resultados esperados que obtiveram classificações maiores que zero e menores que o grau de rigoriedade foram justificados na subseção 6.2.1

Tabelas contendo o alinhamento ao SAFe e ao MR-MPS-SW foram elaboradas para melhor visualização das evidências e da classificação dos processos e resultados esperados. A tabela 11 possui em suas colunas as informações de Nível, Atividade Rotulada do SAFe (AR-SAFE), Evidência da atividade no PDS (EA-PDS) e Classificação do PDS (CL-PDS). Já nas tabelas de alinhamento com o MPS.BR (12 e 13), as colunas possuem as informações de Nível de maturidade, Resultado Esperado, Evidência do resultado esperado no PDS (ERE-PDS) e Classificação do PDS (CL-PDS). Os dados dessa segunda tabela, foram extraídas de [17], que analisou o alinhamento do PDS GAIA e do Processo de Desenvolvimento de Objetos de Aprendizagem da GAIA aos níveis G e F do MPS.BR.

Na subseção 6.2.2, informações extraídas das tabelas são analisadas e interpretadas, possibilitando uma boa avaliação do alinhamento do PDS.

6.1.1 Grau de Rigoriedade

Segundo o grau de rigoriedade proposto por [17], para afirmarmos que um resultado esperado do MPS.BR está implementado em um dos processos do PDS, as evidências devem mostrar que mais de 75% do descrito no resultado está implementado. Essa rigoriedade também é válida para avaliar se um processo do MR-MPS-SW está implementado, utilizando a média dos resultados esperados pertencentes a tal processo, e o nível de maturidade através da média dos processos do nível.

A mesma lógica será aplicada para mensurar se as atividades, eventos e níveis do SAFe estão sendo aplicados.

As métricas que definem as classificações e os correspondentes nominais podem ser vistas na Tabela 10.

Tabela 10 – Classificação quanto à implementação dos processos

Porcentagem (X%)	Nominal
0%	Não Implementado (NI)
$0% < X \leq 25%$	Pouco Implementado (POI)
$25% < X \leq 50%$	Parcialmente Implementado (PAI)
$50% < X \leq 75%$	Implementado (I)
$75% < X \leq 100%$	Totalmente Implementado (TI)

Fonte: Galhardi[17]

6.2 Alinhamento

De acordo com Guia do MR-MPS-SW[5], processos podem ser excluídos do escopo de uma avaliação, ou de uma análise acadêmica como esta, se não forem pertinentes ao negócio da organização. Por isso, o processo de Aquisição (AQU) será excluído da avaliação do PDS pois não é executado pela GAIA.

O alinhamento completo do PDS ao SAFE pode ser visto na Tabela 11, e o alinhamento ao MR-MPS-SW, nas Tabelas 12 e 13.

6.2.1 Justificativas

Nesta subseção são apresentados os motivos que levaram à classificar atividades e resultados esperados como não implementados totalmente, ou seja, cujas implementações encontram-se no intervalo maior que zero e menor que o grau de rigorosidade (75%).

- IPL02: A estimativa e análise de complexidade, dificuldades e tamanho são realizadas, porém para o projeto como um todo e não para cada item do *backlog*;
- EXI01: Não há menção à um controle visual do andamento do projeto;
- EXI02: Há um gerenciamento da comunicação, mas não uma reunião diária como a proposta;
- IRW01: Não há menção explícita se a equipe analisa e discute o *feedback* recebido do cliente;
- IRT01: Recebe-se e analisa-se dados quantitativos e qualitativos, mas não há menção direta sobre uma discussão do que pode ser melhorado na próxima iteração;
- IPI01: Chances de explorar e inovar não estão explícitas no PDS como no SAFE;

- PIP09: Não há menção sobre atribuição de valores de negócio aos objetivos;
- I&A02: A análise de resultados de testes é feita, mas nenhuma medida como a Medida de Previsibilidade do Programa é analisada;
- GFA01: Ocorre o gerenciamento do desenvolvimento, da execução, de cada iteração do projeto e identifica-se riscos e realiza-se correções, mas não há um gerenciamento integrado de todos processos envolvidos em um PI, desde os de planejamento até os de entrega;
- PTF02: Os orçamentos são feitos para o projeto todo, e não para cada incremento. Também não utiliza o método *Lean Budgets*;
- GPR07: Não há menção clara aos recursos humanos em si, apenas a recursos no geral;
- GPP04: O portfólio é monitorado, porém não é especificado se o monitoramento ocorre baseado nos critérios de priorização;
- MED02: O conjunto de medidas é identificado, definido, priorizado e documentado, porém as atualizações, se pertinentes, não são abordadas.

6.2.2 Discussão dos Resultados

Analisando os dados contidos na Tabela 11, percebe-se que das 42 atividades propostas pelo SAFe, 18 encontram-se totalmente implementadas (TI), 7 implementadas (I), 3 parcialmente implementadas (PAI) e 14 não implementadas (NI), a maioria destas são as atividades de *Inspect&Adapt* do nível de Programa.

Substituindo essas classificações nominais pela média dos intervalos que representam, foi possível calcular a média percentual para cada evento do SAFe. Com a média para cada evento calculou-se a média percentual para cada nível, com exceção do nível *Portfolio* que possui apenas um evento, e com a média dos níveis calculou-se a média geral, concluindo que 63% do SAFe está implementado no PDS GAIA.

O gráfico 14 apresenta todas as médias percentuais citadas, sendo possível visualizar os pontos em que o PDS precisa ser mais adaptado. Quanto aos valores apresentados, é importante lembrar que o maior valor possível de ser obtido é 88%, média do intervalo de 76 a 100.

Percebe-se pelo gráfico que o nível *Team* possui a segunda melhor média entre os níveis, e analisando as porcentagens de seus eventos, juntamente com as justificativas apresentadas na subseção anterior, constata-se que para melhorar o alinhamento, o PDS precisa melhorar a comunicação entre os membros da equipe de desenvolvimento.

Tabela 11 – Alinhamento do PDS GAIA ao SAFe

Nível	AR-SAFé	EA-PDS	CL-PDS	Nível	AR-SAFé	EA-PDS	CL-PDS
<i>Team</i>	IPL01		NI	<i>Program</i>	PIP04		NI
	IPL02	PDS-ANI03 PDS-APL09	I		PIP05	PDS-APL01 PDS-APL08 PDS-APL09 PDS-APL11	TI
	IPL03	PDS-APL02 PDS-APL12	TI			PIP06	PDS-APL15
	IPL04	PDS-APL07	TI		PIP07	PDS-ANI05 PDS-APL18	TI
	IPL05	PDS-ANI07	TI			PIP08	PDS-APL16
	IEX01	PDS-EXI01 PDS-EXI02	I		PIP09	PDS-APL08 PDS-APL09	I
		IEX02	PDS-EXI04			PAI	PIP10
	IEX03				NI	PIP11	
		IEX04	PDS-EXI03 PDS-VAT01 PDS-ENT01		TI	PIP12	
	IEX05		PDS-VAT01 PDS-VAT02 PDS-VAT03 PDS-ENT01 PDS-ENT02		TI		PIP13
		IEX06	PDS-APL13 PDS-VAT01		TI	PIP14	
	BRE01		PDS-APL03 PDS-MRQ02 PDS-MRQ03 PDS-MRQ04		TI		ADS01
		IRW01	PDS-ENT05		I	I&A01	
	IRT01		PDS-VAT02 PDS-ENT02		I		I&A02
		IPI01	PDS-VAT01 PDS-ENT01 PDS-MRQ04		PAI	I&A03	
	PIP01				NI		I&A04
		PIP02	PDS-ANI02		TI	I&A05	
PIP03			NI	I&A06			NI
	<i>Program</i>				I&A07		NI
			I&A08			NI	
				GFA01	PDS-EXI02	PAI	
<i>Portfolio</i>			PTF01	PDS-GPT01	TI		
			PTF02	PDS-GPT01	I		
			PTF03	PDS-GPT02	TI		

Fonte: Autor

Tabela 12 – Alinhamento do PDS GAIA ao MR-MPS-SW, parte 1

Nível	Resultado Esperado	ERE-PDS	CL-PDS	Nível	Resultado Esperado	ERE-PDS	CL-PDS
G	GPR01	PDS-ANI02 PDS-APL02	TI	G	GRE01	PDS-MRQ01 PDS-APL01 PDS-APL04	TI
	GPR02	PDS-ANI03 PDS-APL08 PDS-APL10 PDS-APL12	TI		GRE02	PDS-MRQ02 PDS-APL01 PDS-APL03 PDS-APL04	TI
	GPR03	PDS-APL02 PDS-APL05 PDS-APL17	TI		GRE03	PDS-MRQ02 PDS-APL06	TI
	GPR04	PDS-ANI03 PDS-APL10	TI		GRE04	PDS-ENT01 PDS-MRQ04 PDSAPL03	TI
	GPR05	PDS-APL08 PDS-APL10 PDS-APL11	TI		GRE05	PDS-MRQ02 PDS-MRQ03	TI
	GPR06	PDS-APL09	TI	F	GCO01	PDS-APL14 PDS-APL16	TI
	GPR07	PDS-APL08	I		GCO02	PDS-APL14	TI
	GPR08	PDS-ANI04 PDS-APL08	TI		GCO03	PDS-APL14	TI
	GPR09	PDS-APL02 PDS-APL16	TI		GCO04	PDS-EXI01 PDS-EXI05	TI
	GPR10	PDS-ENT05 PDS-APL15	TI		GCO05	PDS-APL14	TI
	GPR11	PDS-ANI05 PDS-APL05 PDS-APL18	TI		GCO06	PDS-APL14	TI
	GPR12	PDS-ANI07 PDS-APL15	TI		GCO07	PDS-EXI05	TI
	GPR13	PDS-EXI02	TI		GQA01	PDS-VAT01 PDS-APL15 PDS-EXI02	TI
	GPR14	PDS-ANI01 PDS-ANI07 PDS-ENT05 PDS-FIN01 PDS-EXI04 PDS-EXI02	TI		GQA02	PDS-VAT01 PDS-APL15 PDS-EXI02	TI
	GPR15	PDS-EXI02 PDS-ENT05 PDS-FIN01	TI		GQA03	PDS-VAT02 PDS-ENT02 PDS-EXI02 PDS-EXI03 PDS-EXI04	TI
	GPR16	PDS-EXI02 PDS-EXI04	TI		GQA04	PDS-VAT03 PDS-ENT03 PDS-EXI02 PDS-EXI03	TI
	GPR17	PDS-EXI02 PDS-APL09	TI		GPP01	PDS-GPT01	TI
	GPR18		NI		GPP02	PDS-GPT02	TI
	GPR19		NI		GPP03	PDS-GPT01	TI
			GPP04	PDS-GPT02	I		
			GPP05	PDS-GPT02	TI		

Fonte: Galhardi[17]

Tabela 13 – Alinhamento do PDS GAIA ao MR-MPS-SW, parte 2

Nível	Resultado Esperado	ERE-PDS	CL-PDS	Nível	Resultado Esperado	ERE-PDS	CL-PDS
F	GPP06		NI	F	MED04	PDS-APL07	TI
	GPP07	PDS-GPT02	TI		MED05	PDS-EXI03	TI
	GPP08	PDS-GPT02 PDS-EXI04	TI		MED06	PDS-EXI03 PDS-EXI04	TI
	MED01	PDS-APL07	TI		MED07	PDS-GPT01 PDS-GPT02	TI
	MED02	PDS-APL07	I				
	MED03	PDS-APL07	TI				

Fonte: Galhardi[17]

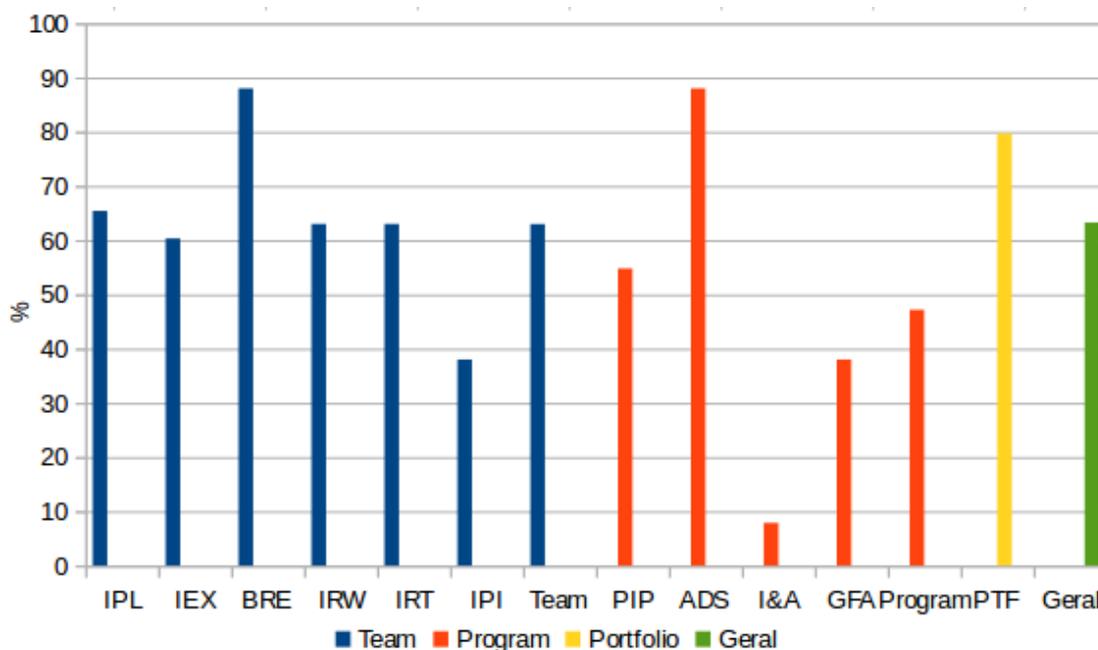


Figura 14 – Gráfico das porcentagens do alinhamento do PDS ao SAFe (Fonte: Autor)

O nível *Program* possui uma porcentagem baixa de alinhamento, sendo o evento de *Inspect & Adapt* um grande responsável por isso. Esse evento prevê a realização de *workshops* para que problemas e erros sejam encontrados e discutidos e não voltem a ser cometidos nas próximas iterações, não havendo nada parecido nas especificações do PDS, o que levou a classificar a maior parte das atividades desse evento como não implementado. Outra classificação que diminuiu a média do nível foi o gerenciamento do trem de desenvolvimento (ART), um conceito novo para o PDS, sendo proposto o uso do Kanban para que possa ser visualizado o andamento do desenvolvimento do trem como um todo.

Já o nível de *Portfolio* possui uma alta porcentagem de alinhamento, sendo a utilização do *Lean Budgets* a única mudança necessária para a completa adesão do PDS à esse nível.

O gráfico 15 apresenta as porcentagens médias de cada resultado esperado, a média de cada nível e a média geral, aplicando a mesma lógica do gráfico anterior.

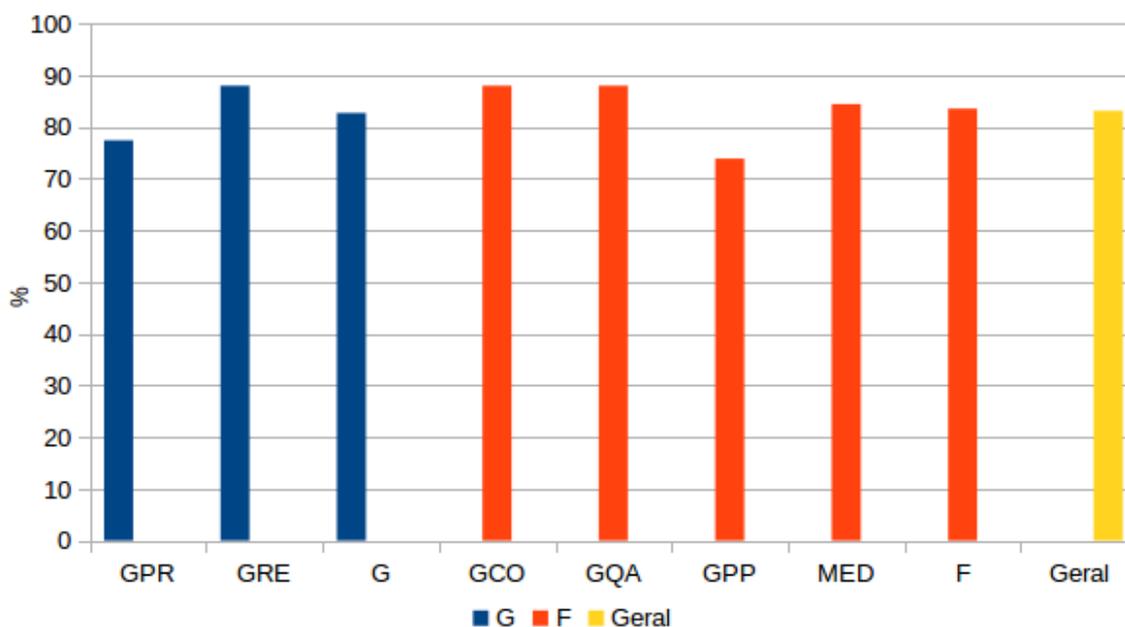


Figura 15 – Gráfico das porcentagens do alinhamento do PDS ao MPS (Fonte: Autor)

Analisando o gráfico percebe-se que o PDS implementa a maioria dos resultados esperados, tanto os do nível G quanto o do F, obtendo uma média geral de 83%, sendo considerado totalmente implementado pelo grau de rigorosidade aplicado. Nota-se ainda que os processos de Gerência de Requisitos (GRE), Gerência de Configuração (GCO) e Gerência de Qualidade (GQA) atingem o valor máximo de 88%, ou seja, todos os seus resultados esperados são implementados pelo PDS.

Comparando-se os resultados obtidos em ambos os alinhamentos, percebe-se que o PDS obtém uma boa adesão aos níveis F e G do modelo de referência MR.MPS, implementando totalmente quase todos os processos, com exceção da Gerência de Portfólio de Projetos (GPP), garantindo qualidade tanto aos produtos e serviços oferecidos quanto ao próprio PDS. Já o alinhamento com a metodologia ágil não obteve uma porcentagem geral tão satisfatória, considerando o SAFe apenas como implementado.

De forma geral, o PDS implementa a maioria dos eventos da metodologia ágil, encontrando dificuldade nos conceitos novos e muito diferentes dos já aplicados. Além disso, uma questão bastante valorizada pelo SAFe e que precisa ser melhorada no PDS é a comunicação entre todos os envolvidos, tanto a verbal quanto a visual, o que proporciona maior envolvimento dos colaboradores e, conseqüentemente, um melhor andamento do projeto.

Um ponto que se manteve totalmente implementado nos dois alinhamentos foi a

gerência de requisitos, sendo considerado bem organizado e refinado por ambos.

A seguir são discutidas algumas mudanças e incrementos que poderiam ser adotados pelo PDS a fim de aumentar o seu alinhamento a metodologia e ao modelo estudados:

- Procurar identificar e compreender os problemas enfrentados durante o desenvolvimento da iteração, discutindo ações que poderiam ser adotadas para corrigir e evitar que tais problemas se repitam, implementando as ações mais votadas ao final da reunião, seria uma atividade que atenderia aos resultados esperados GPR18 e GPR19 e poderia ser adaptada para utilizar as técnicas propostas pelo *Inspect & Adapt* do SAFe;
- Adicionar duas reuniões ao PDS, que diminuiriam o problema de falta de comunicação apresentado anteriormente e aumentariam a implementação de atividades do nível *Team* do SAFe. Uma das reuniões seria a primeira tarefa a ser realizada na etapa de Execução e Implementação e discutiria as capacidades e a disponibilidade de cada membro da equipe, como propõe a atividade IPL01, que passaria a ser totalmente implementada. A outra reunião seria realizada ao final da etapa de Entrega, em que após receber o *feedback* do cliente, a equipe o discutiria e analisaria os porquês dos pontos positivos e negativos que receberam, como propõe a atividade IRW01. Com essa análise, a equipe teria parâmetros para avaliar se os objetivos da iteração foram alcançados e o que poderia ser melhorado nas próximas iterações, realizando a IRT01. Além disso, a descrição da tarefa Gerenciar Comunicação (PDS-EXI04) poderia ser modificada para envolver reuniões diárias como sugere o SAFe;
- Aumentar a participação dos executivos e interessados por parte da organização, levaria a uma melhor contextualização do negócio e de seus valores. Isso mostraria como as ações feitas pelas equipes de desenvolvimento afetam diretamente a empresa, pois influenciam em como os clientes a enxergam, valorizando e aumentando o engajamento dos colaboradores com a organização. Dessa forma, a atividade do SAFe PIP01 poderia ser implementada e a PIP09 poderia ser aplicada melhor;
- Na etapa de Análise e Planejamento do PDS, antes de revisar os planos, uma votação na confiança em que cada membro dos times de desenvolvimento tem em realizar os planos estabelecidos poderia ser feita (atividade PIP12 do SAFe), o que reforçaria pontos específicos dos planos que necessitariam ser refeitos ou adaptados. Além disso, ao final dessa etapa de planejamento, o gerente de projeto poderia expor aos demais envolvidos o que melhorou nessa rodada de planejamento em comparação com a anterior e o que ainda pode ser melhorado, implementando a atividade PIP14 do SAFe.

Adotando essas mudanças, o PDS passaria a ser representado como na Figura 16 e a ter mais uma etapa, a de Análise de Problemas, cujo fluxograma é apresentado na Figura 17. Os fluxogramas modificados das fases Análise e Planejamento, Execução e Implementação e Entrega e Finalização podem ser vistos nas Figuras 18, 19 e 20, respectivamente.

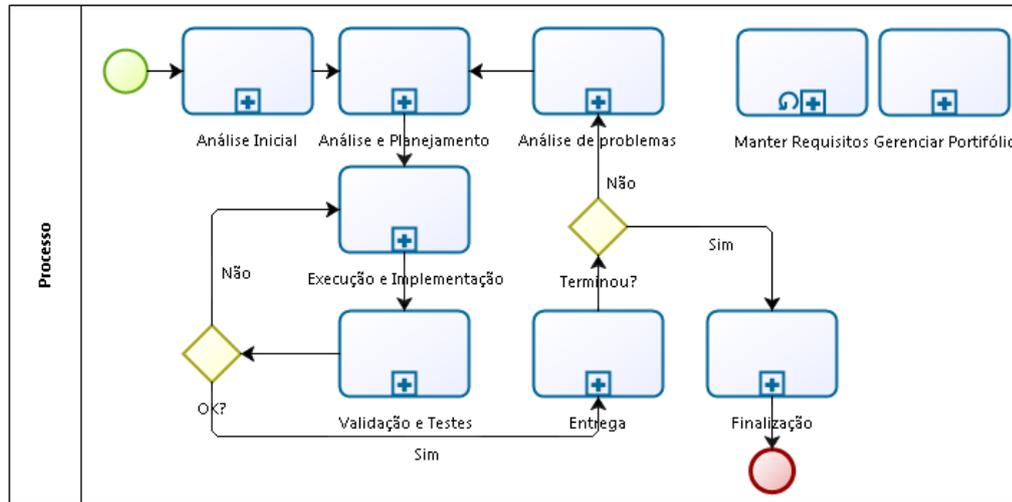


Figura 16 – Processo de Desenvolvimento de *Software* GAIA modificado. (Fonte: Autor)

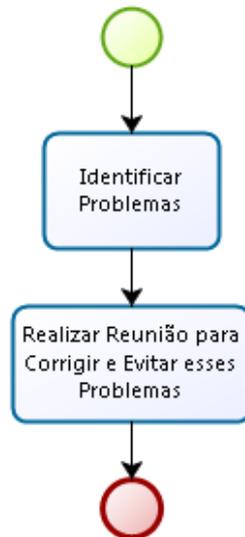


Figura 17 – Análise de Problemas - APR. (Fonte: Autor)

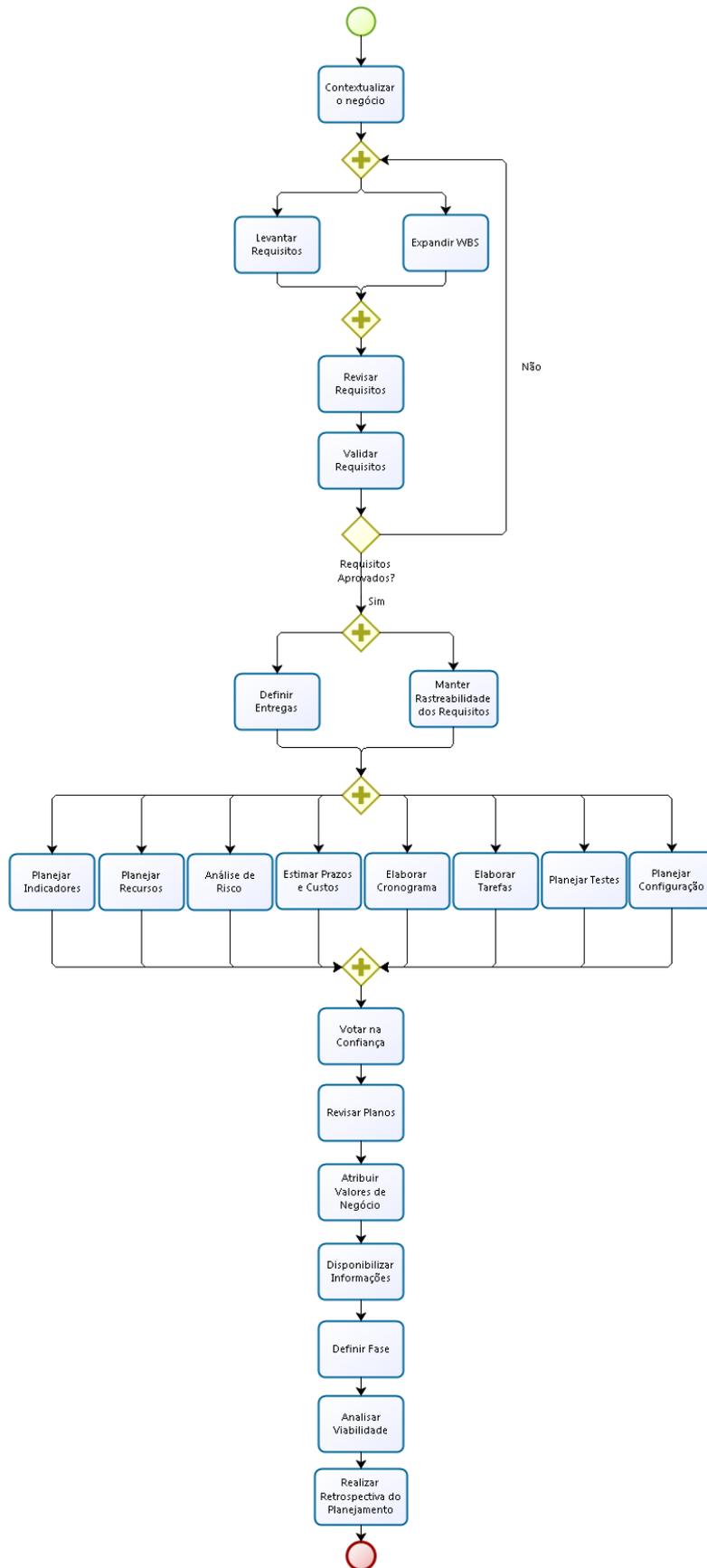


Figura 18 – Análise e Planejamento modificado. (Fonte: Autor)

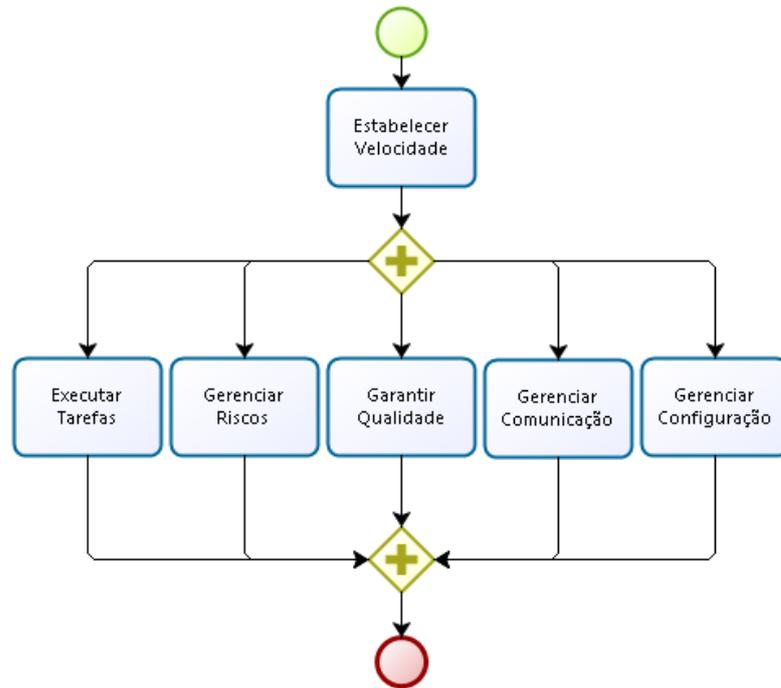


Figura 19 – Execução e Implementação modificado. (Fonte: Autor)

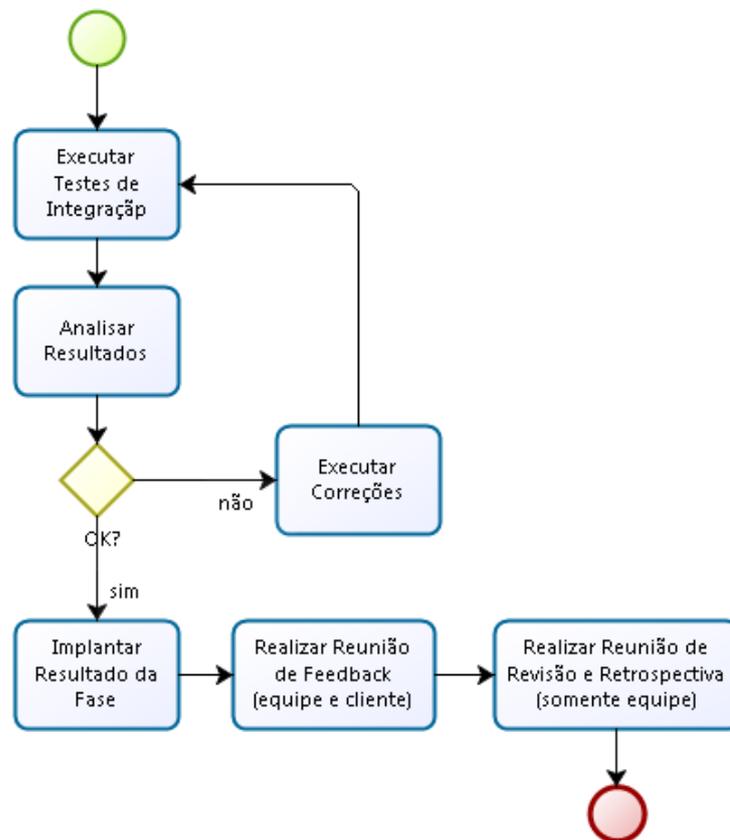


Figura 20 – Entrega modificado. (Fonte: Autor)

7 CONCLUSÃO

Seguindo o comportamento atual do mercado de desenvolvimento de *software*, em que produtos de alta qualidade são exigidos com prazos de entrega cada vez menores, requerendo uma alta produtividade, o Laboratório GAIA procurou desenvolver seu próprio Processo de Desenvolvimento de *Software* para alcançar essas exigências. Espera-se, então, que o PDS seja maduro o suficiente para garantir a qualidade do próprio processo, e siga uma metodologia ágil.

Assim, este trabalho estudou o alinhamento do PDS GAIA ao SAFe, um *framework* que utiliza as metodologias Scrum e XP, além de outras técnicas como Kanban e *Lean Thinking*, para escalonar o uso de metodologias ágeis em equipes de desenvolvimento maiores e até em outros departamentos da organização. Também foi estudado o alinhamento aos níveis G e F do MR-MPS-SW, modelo de referência para *software* que compõe o programa MPS.BR (Melhoria de Processo de *Software* do Brasil).

Através dos alinhamentos foram mapeadas as tarefas do PDS que satisfazem as atividades propostas pelo SAFe e/ou aos resultados esperados do MR-MPS-SW, sendo possível identificar quais melhorias precisam ser realizadas para que o PDS alcance melhores índices de adesão aos modelos estudados.

Por fim, analisando os resultados dos alinhamentos, pode-se concluir que o Processo de Desenvolvimento de *Software* GAIA implementa boa parte do SAFe, precisando melhorar a comunicação entre seus integrantes, e implementa quase todos os resultados esperados dos níveis G e F do MR-MPS-SW.

REFERÊNCIAS

- [1] SOMMERVILLE, I. *Engenharia de software*. 9. ed. [S.l.]: Pearson Prentice Hall, 2011.
- [2] LEFFINGWELL, D. *Scaled Agile Framework*. 2017. Disponível em: <<http://www.scaledagileframework.com>>. Acesso em: 24.07.2017.
- [3] GAIA. *Processo de desenvolvimento de software GAIA*. 2010. Disponível em: <http://gaia.uel.br/projetos/gaia_PDS/default.htm>. Acesso em: 04.09.2017.
- [4] PRESSMAN, R. S. *Engenharia de software: uma abordagem técnica*. 7. ed. [S.l.]: McGraw Hill Education, 2011.
- [5] SOFTEX, A. para promoção da excelência do software brasileiro. Mps.br–melhoria de processo do software brasileiro: Guia geral mps de software. 2016.
- [6] BECK, K. et al. *Manifesto para o desenvolvimento ágil de software*. 2001. Disponível em: <<http://www.manifestoagil.com.br/>>.
- [7] CORTÊS, M. L. *Modelos de qualidade de Software*. 1. ed. [S.l.]: UNICAMP, 2001.
- [8] CMMI Product Team. *Cmmi for development, version 1.2*. 2006. Disponível em: <<https://www.sei.cmu.edu/reports/06tr008.pdf>>.
- [9] International Organization for Standardization and Internacional Electrotechnical Comission. *IEC 12207 systems and software engineering-software life cycle processes. International Organization for Standardization: Geneva*, 2008.
- [10] International Organization for Standardization and Internacional Electrotechnical Comission. *ISO/IEC 15504: Information technology – Process assessment. Part 1: Concepts and vocabulary; Part 2: Performing an assessment; Part 3: Guidance on performind an assessment; Part 4: Guidance on use for process improvement and process capability determination; Part 5: An exemplar process assessment model*. [S.l.], 2003.
- [11] FERREIRA, A. I. F. et al. Applying iso 9001: 2000, mps. br and cmmi to achieve software process maturity: Bl informatica’s pathway. In: IEEE COMPUTER SOCIETY. *Proceedings of the 29th international conference on Software Engineering*. [S.l.], 2007. p. 642–651.
- [12] SILVA, F.; HOENTSCH, S. C.; SILVA, L. Uma análise das metodologias ágeis fdd e scrum sob a perspectiva do modelo de qualidade mps. br. *Scientia Plena*, v. 5, n. 12, 2009.
- [13] ALVES, R. J. F.; STANGA, M. Projetos ágeis xp integrados com o mps. br nível g. *Unoesc & Ciência-ACET*, p. 109–116, 2014.
- [14] PAASIVAARA, M. Adopting safe to scale agile in a globally distributed organization. In: IEEE PRESS. *Proceedings of the 12th International Conference on Global Software Engineering*. [S.l.], 2017. p. 36–40.

- [15] BRENNER, R.; WUNDER, S. Scaled agile framework: Presentation and real world example. In: IEEE. *Software Testing, Verification and Validation Workshops (ICSTW), 2015 IEEE Eighth International Conference on*. [S.l.], 2015. p. 1–2.
- [16] CABRAL, S. P.; FERNANDES, A. M. da R. Teste de softwares utilizando um framework ágil escalável.
- [17] GALHARDI, L. B. *Alinhamento dos Processos de Desenvolvimento de Software do Laboratório GAIA aos níveis G e F do modelo de qualidade MR-MPS-SW*. 2016. Universidade Estadual de Londrina, Londrina - PR. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação).
- [18] SASDELLI, M. C. B. Utilização de ferramentas da qualidade para a geração de inovação em processo: um case de análise de perda em uma indústria de embalagens cartonadas. Universidade Tecnológica Federal do Paraná, 2012.
- [19] NETO, A. C. *Proposta de um modelo de referência para desenvolvimento de software com foco na certificação do MPS. Br*. Tese (Doutorado) — Universidade de São Paulo, 2008.
- [20] COUTO, A. B. *CMMI: integração dos modelos de capacitação e maturidade de sistemas*. [S.l.]: Ciência Moderna, 2007.
- [21] CMMI Product Team. *Cmmi for development (cmmi-dev)*. [S.l.], 2010.
- [22] International Organization for Standardization and International Electrotechnical Commission. ISO/IEC 33020:2015:information technology – process assessment – process measurement framework for assessment of process capability. *International Organization for Standardization: Geneva*, 2015.
- [23] GAIA. *GAIA - Soluções em TIC*. 2017. Disponível em: <www.gaia.uel.br>. Acesso em: 04.09.2017.
- [24] INSTITUTE, P. M. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. 2008.
- [25] ALQUDAH, M.; RAZALI, R. A review of scaling agile methods in large software development. *International Journal on Advanced Science, Engineering and Information Technology*, INSIGHT-Indonesian Society for Knowledge and Human Development, v. 6, n. 6, p. 828–837, 2016.
- [26] RODRIGUES, J. F.; KIRNER, T. G. Benefícios, fatores de sucesso e dificuldades da implantação do modelo mps. br. *IX Simpósio Brasileiro de Qualidade de Software*, p. 41–55, 2010.
- [27] YOSHIDOME, E. et al. Uma apoio sistematizado à implementação do processo de desenvolvimento de requisitos do mps. br e cmmi a partir do uso de ferramentas de software livre. In: *WER*. [S.l.: s.n.], 2012.