

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Requisitos de Software - 201308

Relatório de Projeto

Grupo 05:

André Guedes, Augusto Modesto,
Jonathan Moraes e Matheus Herlan

Orientador:

George Marsicano Corrêa, MSc.

Brasília, DF

02 de Outubro de 2014



André Guedes, Augusto Modesto,
Jonathan Moraes e Matheus Herlan

Relatório de Projeto

Trabalho submetido durante o curso de graduação em Engenharia de Software da Universidade de Brasília como requisito parcial para obtenção curricular da disciplina de Requisitos de *Software*.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: George Marsicano Corrêa, MSc.

Brasília, DF

02 de Outubro de 2014

Sumário

1	INTRODUÇÃO	7
1.1	Visão Geral do Relatório	7
2	CONTEXTO DA EMPRESA (CHAMEX)	9
2.1	Resumo CHAMEX	9
3	JUSTIFICATIVA DA ABORDAGEM	10
4	PROCESSO DE ENGENHARIA DE REQUISITOS	11
4.1	<i>Scaled Agile Framework</i>	12
4.1.1	Nível de Portfólio	13
4.1.2	Nível de Programa	13
4.1.3	Nível de Time	13
4.2	Papéis	14
4.2.1	Nível de Time	14
4.2.2	Nível de Programa	14
4.2.3	Nível de Portfólio	14
4.3	Modelo do Processo de Engenharia de Requisitos	15
4.4	Atividades	16
4.4.1	Nível de Portfólio	16
4.4.1.1	Atividade: Definir Tema(s) de Investimento	16
4.4.1.2	Atividade: Definir Épico(s)	17
4.4.1.3	Atividade: Priorizar Épico(s)	18
4.4.1.4	Atividade: Adotar Épico Priorizado	18
4.4.1.5	Atividade: Gerenciar <i>Backlog</i> de Portfólio	19
4.4.2	Nível de Programa	20
4.4.2.1	Atividade: Definir a Visão	20
4.4.2.2	Atividade: Definir Requisitos Não-Funcionais	21
4.4.2.3	Atividade: Definir <i>Features</i>	22
4.4.2.4	Atividade: Priorizar <i>Features</i>	23
4.4.2.5	Atividade: Adotar <i>Feature</i> Priorizada	24
4.4.2.6	Atividade: Gerenciar <i>Backlog</i> de Programa	24
4.4.3	Nível de Time	25
4.4.3.1	Atividade: Definir Histórias de Usuário	25
4.4.3.2	Atividade: Priorizar Histórias de Usuário	26
4.4.3.3	Atividade: Planejar <i>Sprint</i>	27

4.4.3.4	Atividade: Gerenciar <i>Sprint</i>	28
4.4.3.5	Atividade: Implementar Histórias de Usuário	29
4.4.3.6	Atividade: Desenvolver Protótipo	30
4.4.3.7	Atividade: Realizar Retrospectiva da <i>Sprint</i>	31
4.4.3.8	Atividade: Realizar Revisão da <i>Sprint</i>	32
4.4.3.9	Atividade: Gerenciar <i>Backlog</i> de Time	33
5	ELICITAÇÃO DE REQUISITOS	35
5.1	Técnicas de Elicitação de Requisitos	35
6	TÓPICOS DE GERENCIAMENTO DE REQUISITOS	37
6.1	Rastreabilidade de Requisitos	37
6.1.1	Rastreabilidade Vertical	37
6.1.1.1	Detalhamento dos Atributos para Rastreabilidade	37
6.2	Atributos de Requisitos	38
6.2.1	Prioridade	39
6.2.2	Estabilidade	39
6.2.3	Risco	39
7	PLANEJAMENTO DO PROJETO	41
7.1	Fase de Iniciação	41
7.2	Fase de Planejamento	42
7.3	Fase de Execução e Monitoramento	42
7.4	Cronograma	43
8	FERRAMENTAS DE GESTÃO DE REQUISITOS	53
8.1	Análise de Ferramentas de Gestão de Requisitos	53
8.1.1	A Ferramenta VersionOne	53
8.1.2	A Ferramenta Rally <i>Software</i>	54
8.1.3	Critérios Adotados para Avaliação	55
8.2	Escolha da Ferramenta de Gestão de Requisitos	56
9	CONSIDERAÇÕES FINAIS	57
	Referências	58

Lista de ilustrações

Figura 1 – <i>Scaled Agile Framework</i>	12
Figura 2 – Modelo do Processo de Engenharia de Requisitos	15
Figura 3 – Exemplo de Estratégia de Rastreabilidade Vertical	37
Figura 4 – Ferramenta VersionOne de Gestão de Requisitos	54
Figura 5 – Ferramenta Rally <i>Software</i> de Gestão de Requisitos	55

Lista de tabelas

Tabela 1 – Detalhamento dos Atributos para Rastreabilidade	38
Tabela 2 – Exemplo de Matriz de Atributos	40
Tabela 3 – Tabela de Comparação das Ferramentas	56

Lista de abreviaturas e siglas

CMMI	<i>Capability Maturity Model Integration</i> , Integração do Modelo de Maturidade em Capacitação
IEEE	<i>Institute of Electrical and Electronics Engineers</i> , Instituto de Engenheiros Eletricistas e Eletrônicos
SAFe	<i>Scaled Agile Framework</i> , <i>Framework</i> Ágil Escalável

1 Introdução

A disciplina de Requisitos, de uma maneira geral, agrupa atividades que visam obter o enunciado completo, claro e preciso dos requisitos de um produto de *software*. As seguintes definições de requisitos são aplicáveis e compatíveis com a terminologia do CMMI e dos padrões IEEE (FILHO, 2009):

- Condição ou potencialidade de que um usuário necessita para resolver um problema ou atingir um objetivo;
- Condição ou potencialidade que um sistema, componente ou produto deve possuir para que seja aceito (isto é, satisfaça a um contrato, padrão, especificação ou outro documento formalmente imposto);
- Expressão documentada dessa característica.

É importante ressaltar que os requisitos devem ser levantados pela equipe do projeto, em conjunto com representantes do cliente, usuários e, possivelmente, especialistas da área de aplicação. Ao conjunto de técnicas empregadas para levantar, detalhar, documentar e validar os requisitos de um produto, concretiza a disciplina de Engenharia de Requisitos.

Mediante aos aspectos apresentados até o momento, é importante levar em consideração que os requisitos são especificados de uma maneira particular do ponto de vista da escolha de metodologia (tradicional ou adaptativa). O grupo da disciplina de Requisitos de *Software* adotou a metodologia ágil, baseando-se no *framework* SAFe, proposto por Dean Leffingwell.

1.1 Visão Geral do Relatório

Este relatório está organizado pelos seguintes capítulos:

1. **Introdução:** Descreve conceitos de Requisitos e seu contexto no projeto;
2. **Contexto da Empresa (CHAMEX):** Descreve, resumidamente, o contexto da Empresa fictícia que será contratada pela equipe do projeto para desenvolvimento de uma solução de *software*;
3. **Justificativa da Abordagem:** Descreve os motivos pela decisão de abordagem ágil para o desenvolvimento do projeto;

4. **Processo de Engenharia de Requisitos:** Descreve o processo de Engenharia de Requisitos desenvolvido para o contexto do projeto;
5. **Elicitação de Requisitos:** Descreve a análise e escolha de técnicas de elicitac  o de requisitos para utiliza  o no projeto;
6. **T picos de Gerenciamento de Requisitos:** Fornece dados espec ficos do projeto referentes ao padr o de documento Plano de Gerenciamento de Requisitos, adaptados para o contexto do projeto;
7. **Planejamento do Projeto:** Fornece dados preliminares de planejamento no contexto do projeto;
8. **Ferramentas de Gest o de Requisitos:** Descreve a an lise e escolha de ferramentas de gest o de requisitos para utiliza  o no projeto;
9. **Considera  es Finais:** Conclui o tema do relat rio.

2 Contexto da Empresa (CHAMEX)

O grupo da disciplina de Requisitos de *Software* ficou responsável por trabalhar, juntamente com o grupo da disciplina de Modelagem de Processos, soluções de *software* para o contexto da empresa fictícia CHAMEX. Na Seção 2.1. (Resumo CHAMEX) é descrito brevemente o contexto desta empresa.

2.1 Resumo CHAMEX

1. **Objetivo da Instituição:** Auxiliar pequenas e médias empresas privadas a melhorarem a qualidade de vida dos trabalhadores. Quanto mais disposição, vitalidade e alegria entre os trabalhadores, mais resultado positivo as empresas possuem;
2. **Como:** Através do Modelo de Avaliação (MOA), que busca avaliar o nível de satisfação e qualidade de vida dos trabalhadores de uma empresa. A avaliação das práticas e processos de trabalho são feitas através de questionários.
 - **Questionários:** Os questionários possuem formas e pesos específicos. Suas respostas são controladas e geram um resultado para a empresa com um conjunto de ações necessárias para a melhoria de suas práticas e processos de trabalho. São, ao todo, três questionários:
 - a) Obter a visão da empresa, quanto a suas práticas e processos de trabalho com a alta administração;
 - b) Igualmente ao anterior, contudo, realizado com cada trabalhador da empresa;
 - c) Obter a visão do próprio avaliador sobre o que ele vê na empresa avaliada.

Os resultados da empresa também são comparados com os resultados de outras empresas do mesmo setor, gerando um *ranking* de qualidade de vida de empresas contratantes.

3 Justificativa da Abordagem

O grupo da disciplina de Requisitos de Software elaborou cinco justificativas para adoção da metodologia ágil. Dentre vários aspectos, os tópicos seguintes refletem a justificativa do grupo:

- Experiência por parte de alguns integrantes proveniente de trabalhos extra classe em laboratórios que implementam a metodologia ágil;
- O grupo é disciplinado, contribuindo para maior êxito na utilização da metodologia;
- Concordância com os valores ágeis na questão de documentação, onde documenta-se somente o necessário para o bom andamento do projeto;
- O *framework* SAFe torna transparente as atividades da Engenharia de Requisitos, visíveis em todos os níveis. Além disso, proporciona autonomia para cada gerência;
- A metodologia ágil é mais adequada na interação com o cliente no tocante às mudanças e adaptações.

4 Processo de Engenharia de Requisitos

Como retratado no Capítulo 3 (Justificativa da Abordagem), a abordagem adotada pelo grupo para o desenvolvimento do projeto foi a adaptativa, ou ágil. De forma análoga à abordagem, referenciado por um estudo preciso do SAFe, resolveu-se realizar uma adaptação do *framework*, desenhando um processo próprio de Engenharia de Requisitos.

A seguir, na Seção 4.1 (*Scaled Agile Framework*), é descrito de forma genérica o SAFe, principal referência para a construção do processo de Engenharia de Requisitos proposto.

Na Seção 4.2 (Papéis) é descrito os papéis atribuídos no contexto do projeto.

Na Seção 4.3 (Modelo do Processo de Engenharia de Requisitos) é descrito o modelo do processo de Engenharia de Requisitos proposto para o contexto do projeto.

Na Seção 4.4 (Atividades) é definido minuciosamente cada atividade presente no processo de Engenharia de Requisitos proposto.

4.1 Scaled Agile Framework

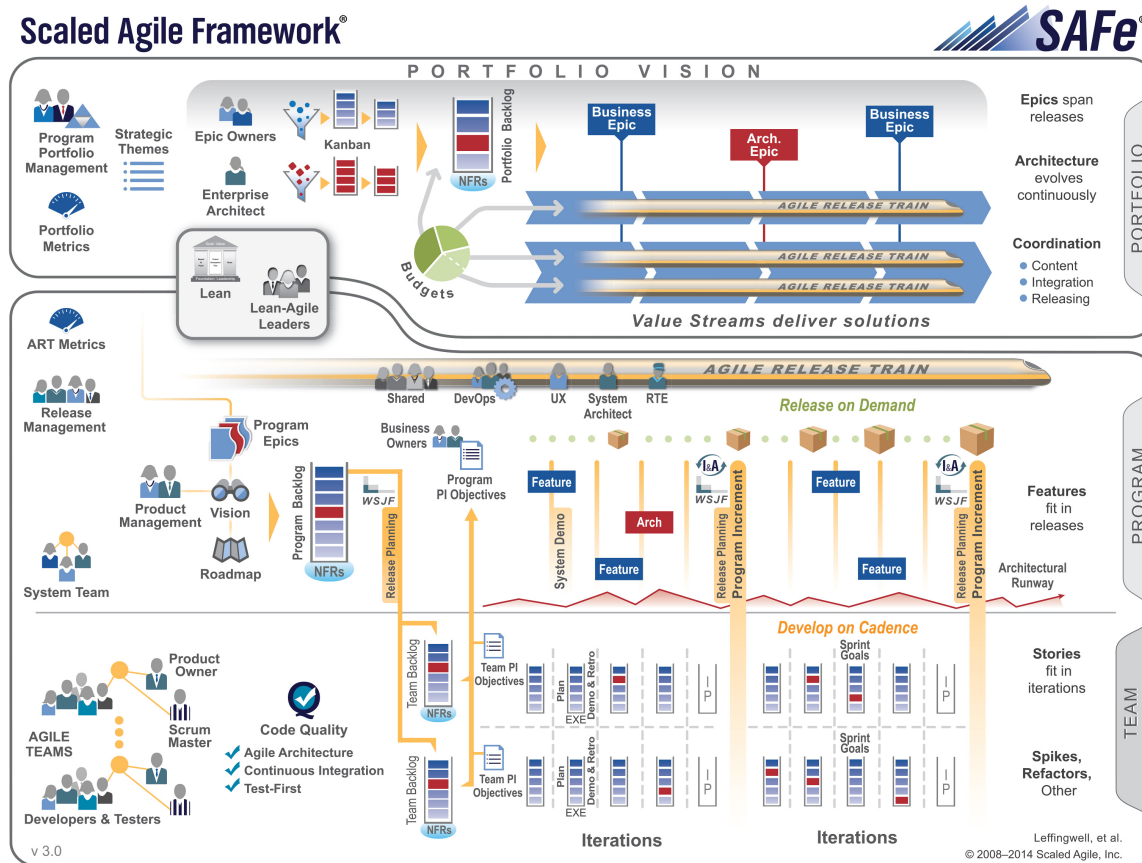


Figura 1 – Visão Geral do SAFe - *Scaled Agile Framework* (LEFFINGWELL, 2011).

4.1.1 Nível de Portfólio

No nível de portfólio, destaca-se os seguintes elementos:

- **Temas de Investimento:** Representam um conjunto de iniciativas que irão dirigir os investimentos, em sistemas, produtos, aplicações e serviços;
- **Épicos:** São iniciativas de desenvolvimento em larga escala que irão gerar a entrega de novos produtos, soluções ou serviços para o cliente;
- **Arquitetura:** São iniciativas arquiteturais que visam dar conta da compatibilização tecnológica entre as soluções atuais e futuras.

4.1.2 Nível de Programa

No nível de programa, destaca-se os seguintes elementos:

- **Roadmap:** Consiste em uma série de releases planejadas, com suas datas, features e prioridades;
- **Features:** Serviços providos pelo sistema que atendem às necessidades dos *Stakeholders*. São itens do *Backlog* do programa;
- **Product Management:** Responsável pela definição das features a nível de programa;
- **Requisitos Não-Funcionais:** Restrições ou condições que devem ser atendidas na construção da solução.

4.1.3 Nível de Time

No nível de time, destaca-se os seguintes elementos:

- **Scrum Master:** Facilita a atuação do time para atingir seus objetivos. Lidera os esforços de melhoria contínua. Faz cumprir regras do processo ágil;
- **Product Owner:** Trabalha com o gerente de produto, analistas de negócio, clientes e outros *Stakeholders* para determinar os requisitos;
- **User Stories:** Pequena declaração de intenção que descreve algo de necessário que o sistema deverá fazer para o usuário;
- **Iterações:** São pequenas, possuem tempo fixo. Entregam incrementos de software que agregam valor ao cliente.

4.2 Papéis

Para o contexto da disciplina de Requisitos de *Software*, levando em consideração a abordagem adaptativa, bem como particularidades do SAFe, os seguintes papéis foram discutidos e atribuídos:

4.2.1 Nível de Time

- ***Product Owner***: Grupo da disciplina de Modelagem de Processos, professor orientador;
- ***Scrum Master***: Monitores das Disciplinas;
- ***Developers & Testers (Desenvolvedores e Testadores de Requisitos)***: Grupo da disciplina de Requisitos de *Software*.

4.2.2 Nível de Programa

- ***Product Management***: Responsabilidade de todo o grupo da disciplina de Requisitos de *Software*.

4.2.3 Nível de Portfólio

- ***Epic Owner***: O papel será executado uma vez pelo time da disciplina de Requisitos de *Software*.

4.3 Modelo do Processo de Engenharia de Requisitos

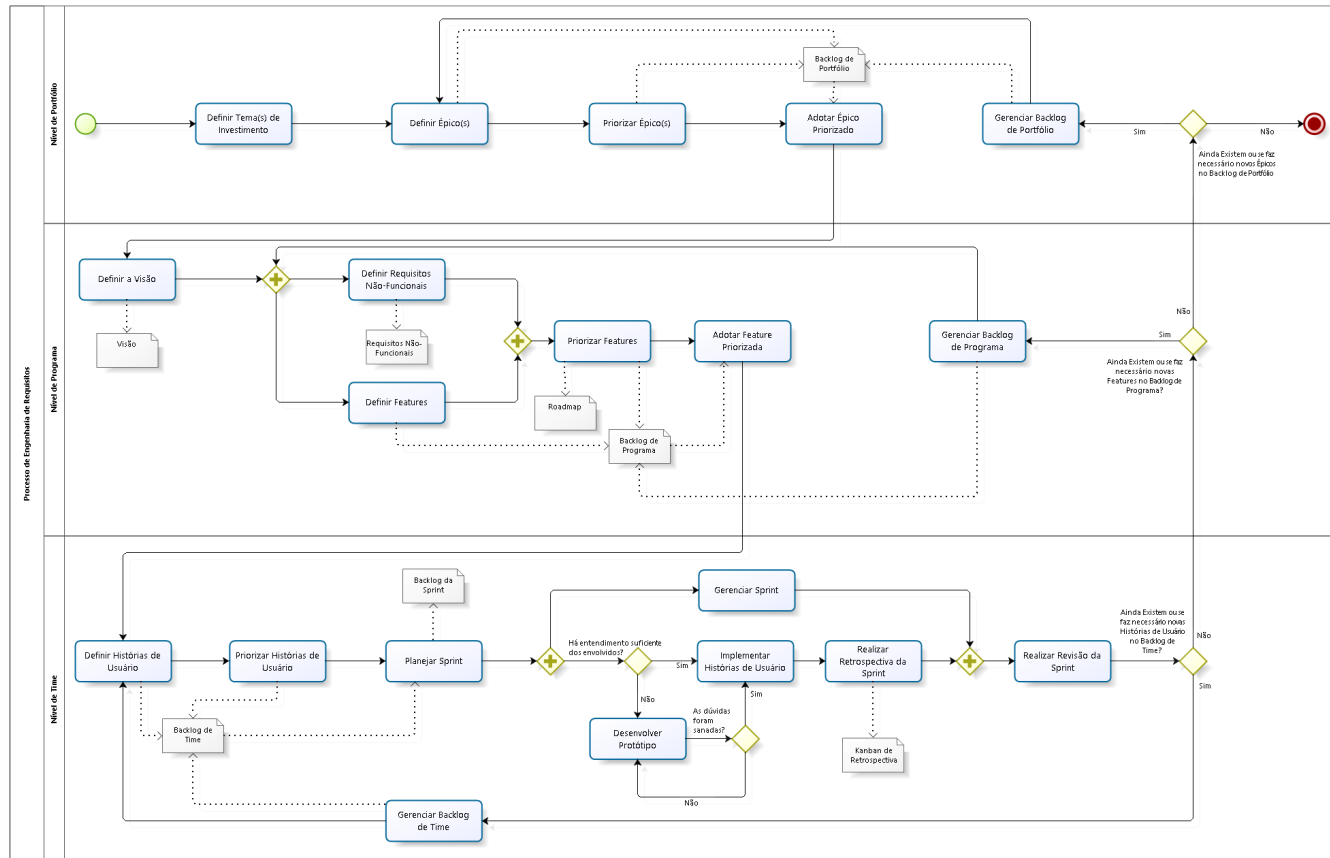


Figura 2 – Modelo do Processo de Engenharia de Requisitos inspirado no SAFe (LEFFINGWELL, 2011).

4.4 Atividades

O Processo de Engenharia de Requisitos descrito no fluxograma da Figura 2 estabelece 3 níveis de atuação, 20 atividades e 7 artefatos principais.

4.4.1 Nível de Portfólio

De acordo com o SAFe (LEFFINGWELL, 2011), o nível de Portfólio é o maior nível do *framework*, onde os programas são alinhados pela estratégia de negócio da organização.

No contexto da disciplina, aborda-se 5 atividades de Portfólio e 1 artefato principal.

4.4.1.1 Atividade: Definir Tema(s) de Investimento

- **Descrição:**

- **SAFe:** Temas de Investimento (*Strategic Themes*, em inglês) são objetivos específicos de negócio que conecta a Visão de Portfólio com a evolução da estratégia de negócio da organização;
- **Contexto do Projeto:** Conceituar itens motrizes da organização e definir Temas de Investimento, que são objetivos específicos de negócio que guiam as prioridades de investimento, assegurando que o trabalho organizado está alinhado com a estratégia de negócio da organização.

- **Tarefa(s):**

1. **Conceituar a Estratégia de Negócio:** Estabelecer a missão e valores da organização;
2. **Conceituar a Análise de Competitivade:** Estabelecer maiores ameaças e áreas de oportunidade para a organização;
3. **Definir Temas de Investimento:** Estabelecer objetivos da organização.

- **Artefato(s) de Entrada:**

- Não há artefatos de entrada para esta atividade.

- **Artefato(s) Gerado(s) e/ou Alterado(s):**

- **Estratégia de Negócio:** Documento descrevendo a missão e valores da organização;
- **Análise de Competitivade:** Documento descrevendo uma análise de ameaças e oportunidades para a organização;
- **Temas de Investimento:** Descrição de mais alto nível da Organização, estabelecendo os objetivos da organização.

- **Atividade(s) Predecessora(s):**

- Esta é a primeira atividade do processo.

- **Atividade(s) Sucessora(s):**

- Atividade: Definir Épico(s).

4.4.1.2 Atividade: Definir Épico(s)

- **Descrição:**

- **SAFe:** Épicos (*Business Epics*, em inglês) são grandes iniciativas, tipicamente transversais ao cliente, que encapsulam novos desenvolvimentos necessários para conquistar benefícios específicos de negócio;
- **Contexto do Projeto:** Elicitar e definir Épicos, que são iniciativas de desenvolvimento que tem como objetivo entregar valor à um tema de investimento.

- **Tarefa(s):**

1. **Elicitar Épico(s) com Stakeholders:** Estabelecer, através do uso de técnicas de elicitação (Ver Capítulo 5), épicos que satisfaçam as intenções de desenvolvimento dos *Stakeholders*;
2. **Validar Épico(s) com Stakeholders:** Verificar se há ambiguidades ou se os épicos não estabelecem uma descrição clara da iniciativa de desenvolvimento. Aplicar modificações necessárias;

- **Artefato(s) de Entrada:**

- **Temas de Investimento:** Descrição de mais alto nível da Organização, estabelecendo os objetivos da organização.

- **Artefato(s) Gerado(s) e/ou Alterado(s):**

- **Épicos:** Descrição de iniciativas de desenvolvimento;
- **Backlog de Portfólio:** Conjunto de Épicos.

- **Atividade(s) Predecessora(s):**

- Atividade: Definir Tema(s) de Investimento.

- **Atividade(s) Sucessora(s):**

- Atividade: Priorizar Épico(s).

4.4.1.3 Atividade: Priorizar Épico(s)

- **Descrição:**

- **SAFe:** Épicos (*Business Epics*, em inglês) são grandes iniciativas, tipicamente transversais ao cliente, que encapsulam novos desenvolvimentos necessários para conquistar benefícios específicos de negócio;
- **Contexto do Projeto:** Ordenar Épicos em relação à sua relevância para o negócio. Épicos são iniciativas de desenvolvimento que tem como objetivo entregar valor à um tema de investimento.

- **Tarefa(s):**

1. **Definir Critérios de Avaliação com *Stakeholders*:** Estabelecer pontos-chave de valores de negócio que serão utilizados para a comparação dos Épicos;
2. **Priorizar Épicos com *Stakeholders*:** Ordenar, de acordo com os critérios estabelecidos, os Épicos do *Backlog* de Portfólio;

- **Artefato(s) de Entrada:**

- ***Backlog* de Portfólio:** Conjunto de Épicos.

- **Artefato(s) Gerado(s) e/ou Alterado(s):**

- ***Backlog* de Portfólio:** Conjunto de Épicos.

- **Atividade(s) Predecessora(s):**

- Atividade: Gerenciar *Backlog* de Portfólio, ou;
- Atividade: Definir Épico(s).

- **Atividade(s) Sucessora(s):**

- Atividade: Adotar Épico Priorizado.

4.4.1.4 Atividade: Adotar Épico Priorizado

- **Descrição:**

- **SAFe:** Épicos (*Business Epics*, em inglês) são grandes iniciativas, tipicamente transversais ao cliente, que encapsulam novos desenvolvimentos necessários para conquistar benefícios específicos de negócio;
- **Contexto do Projeto:** Escolher Épico de maior relevância para o negócio para ser desenvolvido. Épicos são iniciativas de desenvolvimento que tem como objetivo entregar valor à um tema de investimento.

- **Tarefa(s):**
 1. **Confirmar Adoção com *Stakeholders*:** Aprovar a escolha do épico a ser desenvolvido com os *Stakeholders*.
- **Artefato(s) de Entrada:**
 - ***Backlog* de Portfólio:** Conjunto de Épicos;
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - Não há criação nem alteração de artefatos nessa atividade.
- **Atividade(s) Predecessora(s):**
 - Atividade: Priorizar Épico(s).
- **Atividade(s) Sucessora(s):**
 - Atividade: Definir a Visão.

4.4.1.5 Atividade: Gerenciar *Backlog* de Portfólio

- **Descrição:**
 - **SAFe:** *Backlog* de Portfólio é o *Backlog* de maior nível do *framework*, que provê um mecanismo de armazenamento para os Épicos;
 - **Contexto do Projeto:** Gerenciar o *Backlog* de Portfólio, que é um conjunto de épicos armazenados.
- **Tarefa(s):**
 1. **Revisar *Backlog* de Portfólio com *Stakeholders*:** Verificar, com o épico previamente adotado já desenvolvido, se há necessidade de novos épicos além daqueles que já estão no *Backlog* de Portfólio;
- **Artefato(s) de Entrada:**
 - ***Backlog* de Portfólio:** Conjunto de Épicos;
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - ***Backlog* de Portfólio:** Conjunto de Épicos;
- **Atividade(s) Predecessora(s):**

- Atividade: Realizar Revisão da *Sprint*, se:
 - Não existir Histórias de Usuário no *Backlog* de time, e;
 - Não existir *Features* no *Backlog* de Programa, e;
 - Ainda existir Épicos no *Backlog* de Portfólio ou se faz necessário criar novos Épicos.
- **Atividade(s) Sucessora(s):**
 - Atividade: Definir Épico(s).

4.4.2 Nível de Programa

De acordo com o SAFe (LEFFINGWELL, 2011), é no nível de Programa que os esforços dos times ágeis são alinhados e integrados para criar um maior valor às necessidades da organização e dos *Stakeholders*, escalado em três construções principais: Valor, Times e *Timebox*.

No contexto da disciplina, onde possui apenas um time, utiliza-se apenas a construção de Valor, com o objetivo de descrever comportamentos do sistema em um nível alto de abstração através de *Features* e da Visão, além do *Roadmap*. Aborda-se 6 atividades de Programa e 4 artefatos principais.

4.4.2.1 Atividade: Definir a Visão

- **Descrição:**
 - **SAFe:** A Visão descreve um conjunto de soluções à serem desenvolvidas, refletindo as necessidades dos *Stakeholders* e às funcionalidades propostas para atender tais necessidades;
 - **Contexto do Projeto:** Definir a Visão do projeto. A Visão descreve um conjunto de soluções à serem desenvolvidas, refletindo as necessidades dos *Stakeholders* e às funcionalidades propostas para atender tais necessidades.
- **Tarefa(s):**
 1. **Recolher Dados do Nível de Portfólio:** Desenvolver a Visão com dados relativos aos Temas de Investimento e *Backlog* de Portfólio;
 2. **Recolher Dados dos *Stakeholders*:** Desenvolver a Visão com dados adicionais extraídos dos *Stakeholders*;
 3. **Recolher Dados da Organização:** Desenvolver a Visão com dados adicionais extraídos dos integrantes da organização;
 4. **Validar Visão com *Stakeholders*:** Aprovar, com os *Stakeholder*, a Visão.

- **Artefato(s) de Entrada:**

- **Estratégia de Negócio:** Documento descrevendo a missão e valores da organização;
- **Análise de Competitivade:** Documento descrevendo uma análise de ameaças e oportunidades para a organização;
- **Temas de Investimento:** Descrição de mais alto nível da Organização, estabelecendo os objetivos da organização;
- **Backlog de Portfólio:** Conjunto de Épicos;
- **Dados de Stakeholders:** Informações advindas dos *Stakeholders* à serem incrementadas na Visão, em que os *Stakeholders* e/ou a Organização julgaram necessárias;
- **Dados da Organização:** Informações advindas de integrantes da organização à serem incrementadas na Visão, em que os *Stakeholders* e/ou a Organização julgaram necessárias.

- **Artefato(s) Gerado(s) e/ou Alterado(s):**

- **Visão:** Documento que descreve soluções à serem desenvolvidas.

- **Atividade(s) Predecessora(s):**

- Atividade: Adotar Épico Priorizado.

- **Atividade(s) Sucessora(s):**

- Atividade: Definir Requisitos Não-Funcionais, e;
- Atividade: Definir *Features*.

4.4.2.2 Atividade: Definir Requisitos Não-Funcionais

- **Descrição:**

- **SAFe:** Requisitos Não-Funcionais (Nonfunctional requirements, NFRs, em inglês) ou Qualidades do Sistema, descrevem atributos do sistema como: Segurança, confiabilidade, manutenibilidade e usabilidade. Também podem descrever restrições no *design* do sistema;
- **Contexto do Projeto:** Elicitar e definir os Requisitos Não-Funcionais. Os Requisitos Não-Funcionais no contexto equivalem ao SAFe.

- **Tarefa(s):**

1. **Elicitar Requisitos Não-Funcionais com *Stakeholders***: Estabelecer, através do uso de técnicas de elicitação (Ver Capítulo 5), requisitos não-funcionais que satisfaçam as intenções de desenvolvimento dos *Stakeholders*;
 2. **Validar Requisitos Não-Funcionais com *Stakeholders***: Verificar se há ambiguidades ou se os requisitos não-funcionais não estabelecem uma descrição clara. Aplicar modificações necessárias;
- **Artefato(s) de Entrada:**
 - **Visão**: Documento que descreve soluções à serem desenvolvidas.
 - **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - **Requisitos Não-Funcionais**: Atributos e Restrições do Sistema, anexados ao *Backlog* de Portfólio, Programa e Time, comumente denominado de Restrições de *Backlog*.
 - **Atividade(s) Predecessora(s):**
 - Atividade: Definir a Visão, ou;
 - Atividade: Gerenciar *Backlog* de Programa.
 - **Atividade(s) Sucessora(s):**
 - Atividade: Priorizar *Features*, se:
Atividade: Definir *Features* for realizada.

4.4.2.3 Atividade: Definir *Features*

- **Descrição:**
 - **SAFe**: *Features* são funcionalidades do sistema que atendem as necessidades dos *Stakeholders*;
 - **Contexto do Projeto**: Elicitar e definir as *Features*. As *Features* no contexto equivalem ao SAFe.
- **Tarefa(s):**
 1. **Elicitar *Features* com *Stakeholders***: Estabelecer, através do uso de técnicas de elicitação (Ver Capítulo 5), *features* que satisfaçam as intenções de desenvolvimento dos *Stakeholders*;
 2. **Validar *Features* com *Stakeholders***: Verificar se há ambiguidades ou se as *features* não estabelecem uma descrição clara. Aplicar modificações necessárias;
- **Artefato(s) de Entrada:**

- **Backlog de Portfólio:** Conjunto de Épicos.
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - **Features:** Descrição de funcionalidades do sistema;
 - **Backlog de Programa:** Conjunto de *Features*.
- **Atividade(s) Predecessora(s):**
 - Atividade: Definir a Visão, ou;
 - Atividade: Gerenciar *Backlog* de Programa.
- **Atividade(s) Sucessora(s):**
 - Atividade: Priorizar *Features*, se:
Atividade: Definir Requisitos Não-Funcionais for realizada.

4.4.2.4 Atividade: Priorizar *Features*

- **Descrição:**
 - **SAFe:** *Features* são funcionalidades do sistema que atendem as necessidades dos *Stakeholders*;
 - **Contexto do Projeto:** Ordenar *Features* em relação à sua relevância para o desenvolvimento do Épico. As *Features* no contexto equivalem ao SAFe.
- **Tarefa(s):**
 1. **Definir Critérios de Avaliação com *Stakeholders*:** Estabelecer métricas de priorização que serão utilizados para a comparação das *Features*;
 2. **Priorizar *Features* com *Stakeholders*:** Ordenar, de acordo com os critérios estabelecidos, as *Features* do *Backlog* de Programa;
 3. **Desenvolver *Roadmap*:** Estabelecer, em um fluxo temporal, as *Features* à serem desenvolvidas e entregues em forma de *Releases*.
- **Artefato(s) de Entrada:**
 - **Backlog de Programa:** Conjunto de *Features*.
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - **Backlog de Programa:** Conjunto de *Features*;
 - **Roadmap:** Fluxo temporal que descreve a ordem de desenvolvimento das *Features* à serem entregues em forma de *Releases*.

- **Atividade(s) Predecessora(s):**
 - Atividade: Definir Requisitos Não-Funcionais, e;
 - Atividade: Definir *Features*.
- **Atividade(s) Sucessora(s):**
 - Atividade: Adotar *Feature* Priorizada.

4.4.2.5 Atividade: Adotar *Feature* Priorizada

- **Descrição:**
 - **SAFe:** *Features* são funcionalidades do sistema que atendem as necessidades dos *Stakeholders*;
 - **Contexto do Projeto:** Escolher *Feature* de maior relevância para o Épico para ser desenvolvido. As *Features* no contexto equivalem ao SAFe.
- **Tarefa(s):**
 1. **Confirmar Adoção com *Stakeholders*:** Aprovar a escolha da *Feature* a ser desenvolvida com os *Stakeholders*.
- **Artefato(s) de Entrada:**
 - ***Backlog* de Programa:** Conjunto de *Features*.
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - Não há criação nem alteração de artefatos nessa atividade.
- **Atividade(s) Predecessora(s):**
 - Atividade: Priorizar *Features*.
- **Atividade(s) Sucessora(s):**
 - Atividade: Definir Histórias de Usuário.

4.4.2.6 Atividade: Gerenciar *Backlog* de Programa

- **Descrição:**
 - **SAFe:** *Backlog* de Programa é um repositório definitivo para todas as *Features* do projeto;
 - **Contexto do Projeto:** Gerenciar o *Backlog* de Programa, que é um conjunto de *Features* armazenadas.

- **Tarefa(s):**
 1. **Revisar *Backlog* de Programa com *Stakeholders*:** Verificar, com a *Feature* previamente adotada já desenvolvida, se há necessidade de novas *Features* além daqueles que já estão no *Backlog* de Programa;
- **Artefato(s) de Entrada:**
 - ***Backlog* de Programa:** Conjunto de *Features*;
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - ***Backlog* de Programa:** Conjunto de *Features*;
- **Atividade(s) Predecessora(s):**
 - Atividade: Realizar Revisão da *Sprint*, se:
Não existir Histórias de Usuário no *Backlog* de time, e;
Ainda existir *Features* no *Backlog* de Programa ou se faz necessário criar novas *Features*.
- **Atividade(s) Sucessora(s):**
 - Atividade: Definir *Features*.

4.4.3 Nível de Time

De acordo com o SAFe (LEFFINGWELL, 2011), o Nível de Time é uma partição do Nível de Programa, pois não há possibilidades de existir Programa se não over Times. É responsabilidade no Nível de Time a definição, construção e teste de Histórias de Usuário contidas no *Backlog* de Time em uma série de iterações de tempo fixo (também conhecidos como *Sprints*).

No contexto da disciplina, aborda-se 9 atividades de Programa e 3 artefatos principais.

4.4.3.1 Atividade: Definir Histórias de Usuário

- **Descrição:**
 - **SAFe:** Histórias de Usuário (User Stories, em inglês) são descrições detalhadas de uma funcionalidade à ser implementada. Cada história é curta e possui comportamento independente para que seja implementada gradualmente, agregando algum valor ao usuário. Para garantir que toda iteração disponibilize novos valores, histórias são divididos de forma que possam ser completadas em uma única iteração.

- **Contexto do Projeto:** Elicitar e definir as Histórias de Usuário. As Histórias de Usuário no contexto equivalem ao SAFe.
- **Tarefa(s):**
 1. **Elicitar Histórias de Usuário com *Stakeholders*:** Estabelecer, através do uso de técnicas de elicitação (Ver Capítulo 5), Histórias de Usuário que satisfaçam as intenções de desenvolvimento dos *Stakeholders*;
 2. **Validar Histórias de Usuário com *Stakeholders*:** Verificar se há ambiguidades ou se as Histórias de Usuário não estabelecem uma descrição clara. Aplicar modificações necessárias;
- **Artefato(s) de Entrada:**
 - ***Backlog* de Programa:** Conjunto de *Features*;
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - ***Histórias de Usuário*:** Descrições detalhadas de uma funcionalidade do sistema;
 - ***Backlog* de Time:** Conjunto de Histórias de Usuários;
- **Atividade(s) Predecessora(s):**
 - Atividade: Adotar *Feature* Priorizada, ou;
 - Atividade: Gerenciar *Backlog* de Time.
- **Atividade(s) Sucessora(s):**
 - Atividade: Priorizar Histórias de Usuário.

4.4.3.2 Atividade: Priorizar Histórias de Usuário

- **Descrição:**
 - **SAFe:** Histórias de Usuário (*User Stories*) são descrições detalhadas de uma funcionalidade à ser implementada. Cada história é curta e possui comportamento independente para que seja implementada gradualmente, agregando algum valor ao usuário. Para garantir que toda iteração disponibilize novos valores, histórias são divididos de forma que possam ser completadas em uma única iteração.
 - **Contexto do Projeto:** Ordenar Histórias de Usuário em relação à sua relevância para o desenvolvimento da Feature. As Histórias de Usuário no contexto equivalem ao SAFe.

- **Tarefa(s):**
 1. **Definir Critérios de Avaliação com *Stakeholders*:** Estabelecer métricas de priorização que serão utilizados para a comparação das Histórias de Usuário;
 2. **Priorizar Histórias de Usuário com *Stakeholders*:** Ordenar, de acordo com os critérios estabelecidos, as Histórias de Usuário do *Backlog* de Time;
- **Artefato(s) de Entrada:**
 - *Backlog* de Time: Conjunto de Histórias de Usuário.
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - *Backlog* de Time: Conjunto de Histórias de Usuário.
- **Atividade(s) Predecessora(s):**
 - Atividade: Definir Histórias de Usuário.
- **Atividade(s) Sucessora(s):**
 - Atividade: Planejar *Sprint*.

4.4.3.3 Atividade: Planejar *Sprint*

- **Descrição:**
 - **SAFe:** *Sprint* é uma iteração padronizada e simples. De forma simples, uma *Sprint* compreende a definição, construção e validação de incrementos em um período de tempo fixo. A sequência é composta em planejar a iteração, estabelecer um conjunto de Histórias de Usuários e metas, desenvolver as histórias selecionadas, demonstrar o trabalho para os *Stakeholders*, e realizar a retrospectiva da iteração.
 - **Contexto do Projeto:** Planejar a iteração, definindo histórias de usuário à serem implementadas. *Sprint* no contexto equivale ao SAFe.
- **Tarefa(s):**
 1. **Popular o *Backlog* da *Sprint*:** Escolher Histórias de Usuário do *Backlog* de Time para serem implementadas na *Sprint*;
 2. **Definir metas de *Sprint*:** Estabelecer objetivos de implementação da *Sprint*;
 3. **Atribuir Tarefas aos Integrantes do Time:** Dividir as Histórias de Usuário à serem implementadas pelos desenvolvedores do time e atribuir tarefas de gerência aos integrantes.

- **Artefato(s) de Entrada:**
 - **Backlog de Time:** Conjunto de Histórias de Usuário.
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - **Backlog da Sprint:** Subconjunto do *Backlog* de Time, com as Histórias de Usuário a serem implementadas na *Sprint*.
- **Atividade(s) Predecessora(s):**
 - Atividade: Priorizar Histórias de Usuário.
- **Atividade(s) Sucessora(s):**
 - Atividade: Implementar Histórias de Usuário, se:
Há entendimento suficiente dos envolvidos, ou;
 - Atividade: Desenvolver Protótipo, se:
Não há entendimento suficiente dos envolvidos, e;
 - Atividade: Gerenciar *Sprint*.

4.4.3.4 Atividade: Gerenciar *Sprint*

- **Descrição:**
 - **SAFe:** *Sprint* é uma iteração padronizada e simples. De forma simples, uma *Sprint* compreende a definição, construção e validação de incrementos em uma período de tempo fixo. A sequência é composta em planejar a iteração, estabelecer um conjunto de Histórias de Usuários e metas, desenvolver as histórias selecionadas, demonstrar o trabalho para os *Stakeholders*, e realizar a retrospectiva da iteração.
 - **Contexto do Projeto:** Gerenciar a iteração atualizando métricas e respondendo às mudanças. *Sprint* no contexto equivale ao SAFe.
- **Tarefa(s):**
 1. **Gerenciar o *Burndown Chart*:** Criar e manter o gráfico de redução de pontos de Histórias de Usuários da *Sprint*;
 2. **Gerenciar o *Velocity*:** Criar e manter o gráfico de implementação de pontos de Histórias de Usuários da *Sprint*;
 3. **Gerir Time:** Analisar e aperfeiçoar o desenvolvimento do time;
 4. **Responder às Mudanças:** Gerenciar mudanças e acontecimentos inesperados durante a *Sprint*.

- **Artefato(s) de Entrada:**
 - **Backlog da Sprint:** Subconjunto do *Backlog* de Time, com as Histórias de Usuário a serem implementadas na *Sprint*.
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - **Burndown Chart da Sprint:** Gráfico temporal de redução de pontos de Histórias de Usuários contidas no *Backlog* da *Sprint*;
 - **Velocity Chart da Sprint:** Gráfico temporal de implementação de pontos de Histórias de Usuários durante a *Sprint*;
- **Atividade(s) Predecessora(s):**
 - Atividade: Planejar *Sprint*.
- **Atividade(s) Sucessora(s):**
 - Atividade: Realizar Revisão da *Sprint*, se:
Atividade: Realizar Retrospectiva da *Sprint* for realizada;

4.4.3.5 Atividade: Implementar Histórias de Usuário

- **Descrição:**
 - **SAFe:** Histórias de Usuário (*User Stories*) são descrições detalhadas de uma funcionalidade à ser implementada. Cada história é curta e possui comportamente independente para que seja implementada gradualmente, agregando algum valor ao usuário. Para garantir que toda iteração disponibilize novos valores, histórias são divididos de forma que possam ser completadas em uma única iteração.
 - **Contexto do Projeto:** Desenvolver Histórias de Usuário de acordo com os critérios estabelecidos durante o planejamento da *Sprint*. As Histórias de Usuário no contexto equivalem ao SAFe.
- **Tarefa(s):**
 1. **Desenvolver Histórias de Usuário;**
 2. **Verificar Funcionalidades Desenvolvidas:** Apresentar ao time as funcionalidades desenvolvidas, com o objetivo de encontrar falhas e/ou incoerências. Aplicar modificações, se necessário;
 3. **Validar Funcionalidades Desenvolvidas com Stakeholders:** Apresentar ao *Product Owner* e/ou outros *Stakeholders* de relevância, o conjunto de funcionalidades desenvolvido, com o objetivo de validar o desenvolvimento. Aplicar modificações, se necessário;

- **Artefato(s) de Entrada:**
 - **Backlog da Sprint:** Subconjunto do *Backlog* de Time, com as Histórias de Usuário a serem implementadas na *Sprint*.
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - **Incremento Funcional (*Build* ou *Release*):** Conjunto de funcionalidades implementadas durante uma *Sprint*;
- **Atividade(s) Predecessora(s):**
 - Atividade: Planejar *Sprint*, se:
Há entendimento suficiente dos envolvidos, ou;
 - Atividade: Desenvolver Protótipo, se:
As dúvidas foram sanadas.
- **Atividade(s) Sucessora(s):**
 - Atividade: Realizar Retrospectiva da *Sprint*.

4.4.3.6 Atividade: Desenvolver Protótipo

- **Descrição:**
 - **Definição de Prototipar:** Criar um demonstrativo de um sistema novo. Prototipação é essencial para tornar claro informações de requisitos. O *design* do sistema (especificações funcionais) deve ser finalizada antes do sistema ser construída. Enquanto para algumas pessoas pode estar claro a imagem dos requisitos, para outros podem não estar. (PCMAGAZINE, 2014)
 - **Contexto do Projeto:** Desenvolver protótipo(s) com o objetivo de tornar claro a intenção de uma ou mais Histórias de Usuário. As Histórias de Usuário no contexto equivalem ao SAFe.
- **Tarefa(s):**
 1. **Definir Contexto de Prototipação:** Descobrir pontos não compreendidos dos envolvidos e planejar a prototipação dos mesmos;
 2. **Desenvolver Protótipo(s):** Desenvolver, de forma simples, um protótipo funcional que torna claro pontos não compreendidos dos envolvidos;
 3. **Apresentar Protótipo(s):** Apresentar aos envolvidos o(s) protótipo(s) desenvolvido(s). Aplicar modificações, se necessário.
- **Artefato(s) de Entrada:**

- **Backlog da Sprint:** Subconjunto do *Backlog* de Time, com as Histórias de Usuário a serem implementadas na *Sprint*.
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - **Protótipo:** Demonstrativo funcional;
- **Atividade(s) Predecessora(s):**
 - Atividade: Planejar *Sprint*, se:
Não há entendimento suficiente dos envolvidos, ou;
 - Atividade: Desenvolver Protótipo, se:
As dúvidas não foram sanadas.
- **Atividade(s) Sucessora(s):**
 - Atividade: Implementar Histórias de Usuário, se:
As dúvidas foram sanadas.

4.4.3.7 Atividade: Realizar Retrospectiva da *Sprint*

- **Descrição:**
 - **SAFe:** Ao fim de cada iteração, os membros do time se reúnem para a retrospectiva, aonde os membros discutem suas práticas e identifica melhorias. O tempo estabelecido é de uma hora ou menos, cada esforço de retrospectiva é voltado para descobrir **o que ocorreu bem, o que não ocorreu bem** e o que o time pode **fazer melhor da próxima vez**.
- **Tarefa(s):**
 1. **Discutir Aspectos Quantitativos:** Metas, entregas, revisões das métricas de performance;
 2. **Discutir Aspectos Qualitativos:** As práticas do time, desafios enfrentados. Caso problemas sejam identificados, se faz necessário descobrir a causa raiz e a discussão de ações corretivas.
- **Artefato(s) de Entrada:**
 - **Backlog da Sprint:** Subconjunto do *Backlog* de Time, com as Histórias de Usuário a serem implementadas na *Sprint*.
 - **Incremento Funcional (*Build* ou *Release*):** Conjunto de funcionalidades implementadas durante uma *Sprint*;
 - **Pauta da Reunião de Retrospectiva:** Uma agenda simples de pontos à serem discutidos durante a retrospectiva.

- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - **Ata da Retrospectiva:** Resumo da discussão ocorrida na retrospectiva.
- **Atividade(s) Predecessora(s):**
 - Atividade: Implementar Histórias de Usuário.
- **Atividade(s) Sucessora(s):**
 - Atividade: Realizar Revisão da *Sprint*.

4.4.3.8 Atividade: Realizar Revisão da *Sprint*

- **Descrição:**
 - **Revisão da *Sprint* no Contexto do Scrum:** Durante a Revisão da *Sprint*, o time e as partes interessadas colaboram sobre o que acabou de ser feito. Baseados nisso e em mudanças no *Backlog* do Produto feitas durante a *Sprint*, eles colaboram sobre quais são as próximas coisas que podem ser feitas. Essa é uma reunião informal, com a apresentação da funcionalidade, que tem a intenção de promover a colaboração sobre o que fazer em seguida (SCHWABER, 2009);
- **Tarefa(s):**
 1. **Identificar Desenvolvimentos:** Evidenciar o que foi feito e o que não foi feito;
 2. **Apresentar Desenvolvimentos:** Apresentar o conjunto de funcionalidades desenvolvido;
 3. **Discutir Andamento do Projeto:** Estudar *Backlog* de Time no contexto total do projeto, realizar projeções de datas de conclusão prováveis;
 4. **Concluir Revisão da *Sprint*:** Identificar o que foi discutido e estabelecer diretrizes para a próxima *Sprint*.
- **Artefato(s) de Entrada:**
 - ***Backlog* da *Sprint*:** Subconjunto do *Backlog* de Time, com as Histórias de Usuário a serem implementadas na *Sprint*;
 - ***Backlog* de Time:** Conjunto de Histórias de Usuário;
 - ***Backlog* de Programa:** Conjunto de *Features*;
 - **Incremento Funcional (*Build* ou *Release*):** Conjunto de funcionalidades implementadas durante uma *Sprint*;

- **Pauta da Reunião de Revisão de *Sprint*:** Uma agenda simples de pontos à serem discutidos durante a revisão de *Sprint*.
- **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - **Ata da Revisão de *Sprint*:** Resumo da discussão ocorrida na revisão de *Sprint*.
- **Atividade(s) Predecessora(s):**
 - Atividade: Realizar Retrospectiva da *Sprint*.
- **Atividade(s) Sucessora(s):**
 - Atividade: Gerenciar *Backlog* de Time, se:
Ainda existir Histórias de Usuário no *Backlog* de Time ou se faz necessário criar novas Histórias de Usuário, ou;
 - Atividade: Gerenciar *Backlog* de Programa, se:
Não existir Histórias de Usuário no *Backlog* de time, e;
Ainda existir *Features* no *Backlog* de Programa ou se faz necessário criar novas *Features*, ou;
 - Atividade: Gerenciar *Backlog* de Portfólio, se:
Não existir Histórias de Usuário no *Backlog* de time, e;
Não existir *Features* no *Backlog* de Programa, e;
Ainda existir Épicos no *Backlog* de Portfólio ou se faz necessário criar novos Épicos, ou;
 - Evento Determinante de Fim: Encerrar o Processo, se:
Não existir Histórias de Usuário no *Backlog* de time, e;
Não existir *Features* no *Backlog* de Programa, e;
Não existir Épicos no *Backlog* de Portfólio.

4.4.3.9 Atividade: Gerenciar *Backlog* de Time

- **Descrição:**
 - **SAFe:** *Backlog* de Time representa uma coleção de tudo que o time precisa fazer para avançar sua parcela de solução do sistema. Pode conter histórias de usuário, *features* futuras, histórias técnicas, trabalho de infraestrutura, refatorações, e qualquer outra coisa que o time precise fazer;
 - **Contexto do Projeto:** Gerenciar o *Backlog* de Time, que é um conjunto de Histórias de Usuário armazenados.
- **Tarefa(s):**

1. **Revisar *Backlog* de Time com *Stakeholders*:** Verificar, com as Histórias de Usuário previamente adotados já desenvolvidos, se há necessidade de novas Histórias de Usuário além daqueles que já estão no *Backlog* de Time;
 - **Artefato(s) de Entrada:**
 - *Backlog* de Time: Conjunto de Histórias de Usuário.
 - **Artefato(s) Gerado(s) e/ou Alterado(s):**
 - *Backlog* de Time: Conjunto de Histórias de Usuário.
 - **Atividade(s) Predecessora(s):**
 - Atividade: Realizar Revisão da *Sprint*, se:
Ainda existir Histórias de Usuário no *Backlog* de Time ou se faz necessário criar novas Histórias de Usuário.
 - **Atividade(s) Sucessora(s):**
 - Atividade: Definir Histórias de Usuário.

5 Elicitação de Requisitos

A palavra eliciar remete ao significado de descobrimento. Basicamente, eliciar pode ser definido também como tornar explícito, favorecendo a obtenção do máximo de informações para o conhecimento de um determinado objeto em questão.

Em termos gerais, cabe à elicitação a tarefa de identificar os fatos relacionados aos requisitos do sistema, de forma a prover o mais correto e o maior entendimento acerca do que é demandado pelo *software* que está sendo concebido.

Para obtenção de êxito na elicitação de requisitos de um sistema, algumas técnicas foram desenvolvidas. Contudo, deve-se levar em consideração o contexto de trabalho para a contemplação da eficiência da técnica de elicitação. A seguir, na Seção 5.1 (Técnicas de Elicitação de Requisitos), têm-se uma abordagem de algumas técnicas de elicitação.

5.1 Técnicas de Elicitação de Requisitos

As técnicas de elicitação de requisitos têm como propósito a dissipação das dificuldades comuns à fase de levantamento de requisitos. Partindo do princípio que todas as técnicas atualmente utilizadas possuem vantagens e desvantagens, foram definidos critérios para a escolha das técnicas utilizadas pela equipe:

- Adequação com o contexto da equipe (disponibilidade, interatividade e experiência);
- Adequação com a abordagem escolhida;
- Compatibilidade entre as técnicas escolhidas.

Baseando-se nestes critérios, foram avaliadas quatro técnicas de elicitação. A seguir, tem-se uma breve descrição da técnica, bem como uma opinião da equipe da disciplina de Requisitos de *Software* sobre a mesma.

- **Entrevista:** Técnica tradicional para o levantamento de requisitos na fase inicial do processo de obtenção de dados. É uma técnica muito útil quando o contexto permite contato restrito a encontros pontuais. Porém com base nos critérios estabelecidos, esta técnica não se adequa completamente ao contexto da equipe, pois restringe a interatividade e não usufrui da disponibilidade do time como poderia, portanto a entrevista não foi escolhida pela equipe, tendo em vista o que ela pode oferecer com relação às outras;

- **Questionários:** Técnica amplamente aplicada quando a restrição geográfica ou de tempo é uma barreira para encontros presenciais. Geralmente, traz uma visão quantitativa dos resultados e deve haver algum entendimento prévio para a elaboração das questões. Devido a abordagem utilizada, que valoriza a interação entre as pessoas, e o contexto da equipe onde barreiras geográficas e de tempo, a princípio, não seriam grandes problemas, resolveu-se descartar esta técnica como uma escolha possível para as técnicas de elicitação de requisitos que serão usadas no nosso processo de Engenharia de Requisitos;
- **Prototipagem:** Técnica muito utilizada na elicitação de requisitos, pois possibilita uma visão prática, condizente ou não com o produto final baseado no seu nível de fidelidade, que facilita a interpretação concreta dos critérios a serem atingidos para a aceitação da porção da solução na qual a técnica foi utilizada. A escolha desta técnica foi baseada principalmente na adequação à abordagem utilizada, como uma forma de melhor expressar a rastreabilidade dos requisitos através dos protótipos criados, porém, também foi levada em conta a adequação ao contexto da equipe, através da organização das reuniões periódicas da equipe;
- **Brainstorming:** Técnica que consiste em reuniões para a geração de ideias, onde até as ideias não convencionais são encorajadas para a agregação do maior número de ideias possíveis para serem revisadas e escolhidas, favorecendo o surgimento de soluções criativas para o problema. A escolha desta técnica foi baseada no contexto da equipe, pois pode ser realizada tanto presencialmente quanto remotamente. Na abordagem escolhida, torna mais fácil a discussão e interação na elicitação dos requisitos, o que é valorizado nas metodologias ágeis e também na compatibilidade entre as técnicas escolhidas, pois as técnicas se complementam e trazem duas visões diferentes que abrangem amplamente as possíveis dúvidas quanto aos requisitos levantados.

6 Tópicos de Gerenciamento de Requisitos

A abordagem de Gerenciamento de Requisitos está baseada no Plano de Gerenciamento de Requisitos. Adicionalmente, é importante ressaltar que apenas alguns tópicos foram adotados, tais como a Estratégia de Rastreabilidade de Requisitos e os Atributos de Requisitos.

6.1 Rastreabilidade de Requisitos

A estratégia para a rastreabilidade de requisitos começa a partir do nível de portfólio e percorre os demais níveis, de programa e de time, em suas dadas atividades.

6.1.1 Rastreabilidade Vertical

De maneira a obter uma execução concisa do processo de Engenharia de Requisitos desenhado para o contexto da disciplina de Requisitos de *Software*. A Figura 3 indica visualmente a estratégia de rastreabilidade abordada.

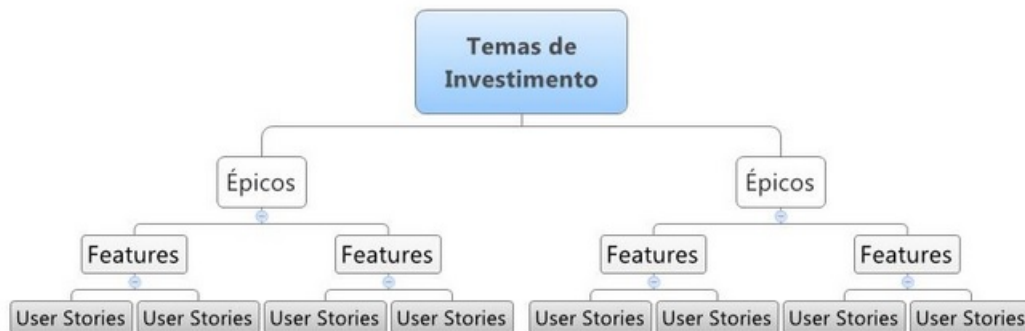


Figura 3 – Exemplificação visual da estratégia de rastreabilidade vertical.

6.1.1.1 Detalhamento dos Atributos para Rastreabilidade

Atributos de Tema de Investimento		
	Valor	Descrição
ID	<i>PT – código_tema_investimento</i>	Identificador do Tema de Investimento - Nível de Portfólio
Atributos de Épicas		
	Valor	Descrição
ID	<i>PT – código_tema_investimento : código_épico</i>	Identificador do Épico - Nível de Portfólio
Data Início	<i>Data</i>	Data planejada de iniciação do épico
Data Encerramento	<i>Data</i>	Data planejada para encerramento do épico
Atributos de Feature		
	Valor	Descrição
ID	<i>PR – código_tema_investimento : código_épico : código_feature</i>	Identificador da Feature existente dentro de um determinado épico - Nível de Programa
Atributos de User Stories		
	Valor	Descrição
ID	<i>T – código_tema_investimento : código_épico : código_feature : código_user_stories</i>	Identificador de um User Story existente dentro de uma determinada Feature - Nível de Time

Tabela 1 – Detalhamento dos atributos para rastreabilidade, vale ressaltar que os códigos são compostos de dois dígitos. Exemplo: T:01:02:04:12.

6.2 Atributos de Requisitos

Atributos de requisitos são conceitos que caracterizam o requisito de acordo com um dado padrão. Os atributos aos quais os requisitos serão relacionados são:

- Prioridade;
- Estabilidade;

- Risco.

Nas subseções a seguir são apresentados a formalização do padrão desses atributos de requisitos.

6.2.1 Prioridade

Este atributo é utilizado para determinar qual a importância do requisito com relação ao funcionamento ideal do sistema. As opções para este atributo são:

- **Essencial:** Indica um requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que devem ser implementados impreterivelmente;
- **Importante:** Indica um requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim;
- **Desejável:** Indica um requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

6.2.2 Estabilidade

Este atributo está interligado com a possibilidade do requisito sofrer modificações ao longo do projeto ou em seu tempo de vida do sistema. As opções para este atributo são:

- **Alta:** Indica um requisito muito estável, ou seja, é pouco provável que o requisito em questão sofra mudanças no decorrer do projeto;
- **Média:** Indica um requisito que tem uma probabilidade média de sofrer alguma mudança;
- **Baixa:** Indica um requisito não estável, ou seja, é muito provável que o requisito em questão necessite de alguma mudança no futuro.

6.2.3 Risco

Relacionado à possibilidade do requisito gerar algum risco capaz de prejudicar ou até mesmo abortar o desenvolvimento do sistema. As opções para este atributo são:

- **Alto:** Indica que os riscos de impactos são grandes e significativos;
- **Médio:** Indica que os riscos de impactos são grandes mas com impactos não tanto significativos;
- **Baixo:** Indica a ausência de riscos de impacto no desenvolvimento do sistema.

Na Tabela 2 é apresentado um pequeno exemplo que torna mais clara a aplicação desses atributos de requisitos às histórias de usuários desenvolvidas.

Requisito	Prioridade	Estabilidade	Risco
Cadastrar Exemplo	Essencial	Alta	Baixo
Remover Exemplo	Importante	Média	Baixo
Atualizar Exemplo	Essencial	Alta	Médio

Tabela 2 – Exemplo de Matriz de Atributos.

7 Planejamento do Projeto

Para que seja possível contemplar uma visão prévia de todo o trabalho que a equipe deve desenvolver, foi construído um cronograma.

O cronograma abrange uma visão genérica de gerenciamento de projetos, exibindo as fases de Iniciação, Planejamento, Execução e Monitoramento. Adicionalmente, dentro de cada fase, várias atividades foram definidas, bem como as datas de início e término das mesmas.

Além dos aspectos supracitados, cada atividade foi atribuída a um determinado integrante da equipe. Este, por sua vez, poderá fazer uma atualização do status de andamento da atividade. Dessa maneira, será possível efetuar um maior controle da execução e possíveis reajustes no decorrer do trabalho.

Dentro do contexto de planejamento do projeto, o nosso ciclo de vida compreende 3 fases à serem desenvolvida em cerca de 3 meses, compondo ao todo 12 iterações.

Na Seção 7.1 é descrito a fase de iniciação.

Na Seção 7.2 é descrito a fase de planejamento.

Na Seção 7.3 é descrito a fase de execução e monitoramento.

Na Seção 7.4 é descrito o cronograma do projeto.

7.1 Fase de Iniciação

- **Descrição:** Nesta fase, contemplam-se as atividades básicas para início do desenvolvimento do projeto proposto pelas disciplinas de Requisitos de Software e Modelagem de Processos. É possível constatar a definição de metodologia por parte da equipe de requisitos (tradicional, adaptativa ou híbrida) bem como a definição em conjunto do cronograma preliminar do projeto;
- **Marco(s) da Fase:** Definição da Abordagem Metodológica de Requisitos, Planejamento do Projeto;
- **Data de Início:** 04 de Setembro;
- **Data de Término:** 11 de Setembro;
- **Duração Total:** 7 Dias;
- **Status:** Concluída;

7.2 Fase de Planejamento

- **Descrição:** Nesta fase, contemplam-se as atividades propostas pela primeira parte do projeto, levando em consideração as particularidades das disciplinas de Requisitos de Software e Modelagem de Processos. As atividades desenvolvidas nessa fase caracterizam-se como subsídios de entrada para as atividades que serão desenvolvidas ao longo da Fase de Execução e Monitoramento;
- **Marco(s) da Fase:** Modelagem do Processo Atual da Empresa (AS-IS), Escolha da Ferramenta BPMS, Relatório de Requisitos;
- **Data de Início:** 12 de Setembro;
- **Data de Término:** 02 de Outubro;
- **Duração Total:** 21 Dias;
- **Status:** Concluída;

7.3 Fase de Execução e Monitoramento

- **Descrição:** Nesta fase, contemplam-se as atividades inerentes à execução do processo de Engenharia de Requisitos definido na Fase de Planejamento. Adicionalmente, constata-se atividades que correlacionam os resultados obtidos no planejamento por parte das disciplinas de Engenharia de Requisitos e Modelagem de Processos, trabalhando o contexto de negócio proposto juntamente com a aplicação das técnicas e metodologias ensinadas em sala de aula;
- **Marco(s) da Fase:** Soluções Incrementais de Requisitos, Soluções de Negócio de Modelagem de Processos;
- **Data de Início:** 03 de Outubro;
- **Data de Término:** 06 de Dezembro;
- **Duração Total:** 64 Dias;
- **Status:** Em Execução;

7.4 Cronograma

Fase de Iniciação: Iteração Única						
Disciplina de Requisitos de Software						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Definição da Abordagem Metodológica	01	04/09		Requisitos	Ata da Definição	100
Disciplina de Requisitos de Software & Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Reunião para Definição do Cronograma	01	09/09		Todos os Grupos	Ata de Reunião	100
Entrega do Planejamento do Projeto	01	11/09		Todos os Grupos	Cronograma	100
Fase de Planejamento: Iteração 01						
Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Modelagem do Processo Atual da Empresa (AS-IS)	05	16/09	20/09	Modelagem	Documento AS-IS	100
Disciplina de Requisitos de Software						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Avaliar e Selecionar uma Ferramenta para Gestão de Requisitos	10	11/09	20/09	Requisitos	Detalhamento da Escolha da Ferramenta	100

Definir Processo de Engenharia de Requisitos	10	11/09	20/09	Requisitos	Detalhamento de Atividades, Responsabilidades e Papéis no Processo	100
Disciplina de Requisitos de Software & Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Avaliar e Selecionar uma Ferramenta de BPMS	09	11/09	19/09	Todos os Grupos	Detalhamento da Escolha da Ferramenta	100
Reunião para Formalização da Escolha da Ferramenta de BPMS	01	20/09		Todos os Grupos	Ata de Reunião	100
Fase de Planejamento: Iteração 02						
Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	Concl.
Modelagem do Processo Atual da Empresa (AS-IS)	02	21/09	22/09	Modelagem	Documento AS-IS	100
Realizar Simulação (AS-IS)	01	22/09		Modelagem	Simulação AS-IS	100
Avaliar Resultado do Processo	01	22/09		Modelagem	Análise dos Resultados	100
Apresentar Resultados (AS-IS)	01	22/09		Modelagem	Documento de Resultados	100
Descrever Atividades e Regras de Negócio (AS-IS)	04	22/09	25/09	Modelagem	Documento de Atividades e Regras de Negócio da Empresa	100
Identificar os Pontos de Automa-tização e Melhoria de Processos	04	22/09	25/09	Modelagem	Pontos de Automa-tização e Me-lhoria de Processos Documenta-dos	100
Refinar Modelos (AS-IS)	04	22/09	25/09	Modelagem	Modelo AS-IS Refinado	100

Disciplina de Requisitos de Software						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Avaliar e Selecionar uma Ferramenta para Gestão de Requisitos	04	21/09	24/09	Requisitos	Detalhamento da Escolha da Ferramenta	100
Desenhar Modelo de Processo de Engenharia de Requisitos na Ferramenta de BMPS	02	21/09	22/09	Requisitos	Desenho do Processo de Engenharia de Requisitos	100
Workshop para Validação do Processo de Engenharia de Requisitos	01	23/09		Requisitos	Diretrizes e Propostas de Aprimoramento para o Processo	100
Ponto de Controle 01	01	25/09		Requisitos	Formalização e Revalidação do Processo de Engenharia de Requisitos	100
Fase de Planejamento: Iteração 03						
Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	Concl.
Descrever Atividades e Regras de Negócio (AS-IS)	04	26/09	29/09	Modelagem	Documento de Atividades e Regras de Negócio da Empresa	100
Identificar os Pontos de Automa-tização e Melhoria de Processos	04	26/09	29/09	Modelagem	Pontos de Automatização e Melhoria de Processos Documentados	100
Refinar Modelos (AS-IS)	04	26/09	29/09	Modelagem	Modelo AS-IS Refinado	100
Disciplina de Requisitos de Software & Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.

Disciplina de Requisitos de Software						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Estabelecer Técnicas de Elicitação	03	25/09	27/09	Requisitos	Detalhamento das Técnicas de Elicitação de Requisitos	100
Elaborar Plano de Gerenciamento de Requisitos	04	25/09	28/09	Requisitos	Plano de Gerenciamento de Requisitos	100
Apresentação 01	01	02/10		Requisitos		100
Disciplina de Requisitos de Software & Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Reunião de Apresentação AS-IS	01	29/09		Todos os Grupos	Ata de Reunião	100
Fase de Execução e Monitoramento: Iteração 01						
Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	Concl.
Refinamento da Identificação Inicial	07	06/10	12/10	Modelagem		100
Modelagem do Processo Proposto (TO-BE)	18	03/10	20/10	Modelagem		100
Documentar Melhorias (TO-BE)	18	03/10	20/10	Modelagem		100
Definir Metas e Indicadores (TO-BE)	18	03/10	20/10	Modelagem		100
Validar as Melhorias (TO-BE)	18	03/10	20/10	Modelagem		100
Planejar a Implementação do Processo (TO-BE)	18	03/10	20/10	Modelagem		100

Disciplina de Requisitos de Software & Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Formalização do Entendimento do Contexto de Negócio	03	03/10	05/10	Todos os Grupos		100
Geração de Solução de BPMS	07	14/10	20/10	Todos os Grupos	Solução Automatizada	100
Fase de Execução e Monitoramento: Iteração 02						
Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	Concl.
Modelagem do Processo Proposto (TO-BE)	20	21/10	10/11	Modelagem		100
Documentar Melhorias (TO-BE)	20	21/10	10/11	Modelagem		100
Definir Metas e Indicadores (TO-BE)	20	21/10	10/11	Modelagem		100
Validar as Melhorias (TO-BE)	20	21/10	10/11	Modelagem		100
Planejar a Implementação do Processo (TO-BE)	20	21/10	10/11	Modelagem		100
Disciplina de Requisitos de Software						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Registrar os Requisitos e a Rastreabilidade na Ferramenta	20	21/10	10/11	Requisitos	Registro dos Requisitos na Ferramenta de Gerência de Requisitos	0
Definir Tema(s) de Investimento	10	21/10	30/10	Matheus Ferraz	Estratégias de Negócio, Análise de Competitivade, Temas de Investimento	0

Definir Épicas(s)	08	23/10	30/10	Matheus Ferraz	Épicas, Backlog de Portfólio	0
Priorizar Épicas(s)	08	23/10	30/10	Caio Nardelli	Backlog de Portfólio	0
Adotar Épicas Priorizado	01	30/10		Caio Nardelli		0
Definir Visão	10	23/10	01/11	Jonathan Moraes	Visão	0
Definir Requisitos Não-Funcionais	10	26/10	04/11	André Guedes	Requisitos Não-Funcionais	0
Definir Features	10	26/10	04/11	André Guedes	Features, Backlog de Programa	0
Priorizar Features	01	04/11		Jonathan Moraes	Backlog de Programa	0
Adotar Feature Priorizada	01	04/11		Jonathan Moraes		0
Definir Histórias de Usuário	07	04/11	10/11	Caio Nardelli	Histórias de Usuário, Backlog de Time	0
Priorizar Histórias de Usuário	01	04/11	10/11	Caio Nardelli	Backlog de Time	0
Disciplina de Requisitos de Software & Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Geração de Solução de BPMS	20	21/10	10/11	Todos os Grupos	Solução Automatizada	0
Fase de Execução e Monitoramento: Iteração 03 (Sprint 0)						
Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	Concl.
Realizar Simulação do Processo (TO-BE)	18	11/11	17/11	Modelagem		0
Avaliar Resultado do Processo (TO-BE)	07	11/11	17/11	Modelagem		0
Comparar e Analisar os Resultados do Processo Redesenhado (TO-BE) e o Processo Atual (AS-IS)	18	11/11	17/11	Modelagem		0
Monitorar o Processo de Melhoria	07	11/11	17/11	Modelagem		0

Planejar a Implementação do Processo (TO-BE)	07	11/11	17/11	Modelagem		0
Monitorar o Processo de Melhoria (TO-BE)	01	17/11		Modelagem		0
Disciplina de Requisitos de Software						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Registrar os Requisitos e a Rastreabilidade na Ferramenta	07	11/11	17/11	Requisitos	Registro dos Requisitos na Ferramenta de Gerência de Requisitos	0
Planejar Sprint	01	11/11		Matheus Ferraz	Backlog da Sprint	0
Gerenciar Sprint	06	11/11	16/11	André Guedes	Burndown Chart da Sprint, Velocity Chart da Sprint	0
Desenvolver Histórias de Usuário	05	11/11	15/11	André Guedes	Protótipos, Incremento Funcional	0
Retrospectiva da Sprint	01	16/11		Jonathan Moraes	Ata da Retrospectiva de Sprint	0
Revisão da Sprint	01	17/11		Caio Nardelli	Ata da Revisão de Sprint	0
Disciplina de Requisitos de Software & Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Geração de Solução de BPMS	07	11/11	17/11	Todos os Grupos	Solução Automatizada	0
Fase de Execução e Monitoramento: Iteração 04 (Sprint 1)						
Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	Concl.
Realizar Simulação do Processo (TO-BE)	07	18/11	24/11	Modelagem		0
Avaliar Resultados do Processo (TO-BE)	07	18/11	24/11	Modelagem		0

Comparar e Analisar os Resultados do Processo Redesenhado (TO-BE) e o Processo Atual (AS-IS)	07	18/11	24/11	Modelagem		0
Disciplina de Requisitos de Software						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Registrar os Requisitos e a Rastreabilidade na Ferramenta	07	18/11	24/11	Requisitos	Registro dos Requisitos na Ferramenta de Gerência de Requisitos	0
Gerenciar Backlogs	01	18/11		Caio Nardelli	Backlogs Relacionados e Requisitos	0
Desenvolver Requisitos	06	19/11	24/11	Jonathan Moraes	Artefatos Gerais de Requisitos	0
Disciplina de Requisitos de Software & Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Geração de Solução de BPMS	07	18/11	24/11	Todos os Grupos	Solução Automatizada	0
Fase de Execução e Monitoramento: Iteração 05 (Sprint 2)						
Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	Concl.
Realizar Simulação do Processo (TO-BE)	07	25/11	01/12	Modelagem		0
Avaliar Resultados do Processo (TO-BE)	07	25/11	01/12	Modelagem		0
Comparar e Analisar os Resultados do Processo Redesenhado (TO-BE) e o Processo Atual (AS-IS)	07	25/11	01/12	Modelagem		0

Disciplina de Requisitos de Software						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Registrar os Requisitos e a Rastreabilidade na Ferramenta	07	25/11	01/12	Requisitos	Registro dos Requisitos na Ferramenta de Gerência de Requisitos	0
Gerenciar Backlogs	01	25/11		André Guedes	Backlogs Relacionados e Requisitos	0
Desenvolver Requisitos	06	26/11	01/12	Matheus Ferraz	Artefatos Gerais de Requisitos	0
Disciplina de Requisitos de Software & Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Geração de Solução de BPMS	07	25/11	01/12	Todos os Grupos	Solução Automatizada	0
Fase de Execução e Monitoramento: Iteração Final						
Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	Concl.
Realizar Simulação do Processo (TO-BE)	02	02/12	03/12	Modelagem		0
Avaliar Resultados do Processo (TO-BE)	02	02/12	03/12	Modelagem		0
Comparar e Analisar os Resultados do Processo Redesenhado (TO-BE) e o Processo Atual (AS-IS)	02	02/12	03/12	Modelagem		0
Disciplina de Requisitos de Software						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Registrar os Requisitos e a Rastreabilidade na Ferramenta	02	02/12	03/11	Requisitos	Registro dos Requisitos na Ferramenta de Gerência de Requisitos	0

Disciplina de Requisitos de Software & Disciplina de Modelagem de Processos						
Atividade	Dias	Início	Fim	Responsável	Resultado(s)	% Concl.
Geração de Solução de BPMS	02	02/12	03/12	Todos os Grupos	Solução Automatizada	0
Formalização dos Resultados	02	03/12	04/12	Todos os Grupos	Análise dos Acontecimentos	0
Apresentação Final	01	06/12		Todos os Grupos		0

8 Ferramentas de Gestão de Requisitos

Para um bom desenvolvimento do trabalho, foram avaliadas duas ferramentas para Gerência de Requisitos: VersionOne e Rally *Software*.

A ferramenta VersionOne foi indicada pelo grupo de monitores da equipe e a ferramenta Rally *Software* foi descoberta em um artigo sobre discussão de metodologias ágeis. É importante ressaltar que as duas ferramentas lidam com contextos ágeis.

8.1 Análise de Ferramentas de Gestão de Requisitos

Nas subseções à seguir são descritas uma breve abordagem de cada ferramenta e os critérios adotados para avaliação.

8.1.1 A Ferramenta VersionOne

VersionOne é uma ferramenta de suporte ao desenvolvimento de *software* utilizando metodologias ágeis, tais como Scrum, *eXtreme Programming*, ou mesmo metodologias híbridas. De maneira geral, foi concebida para equipes pequenas de forma a promover a melhoria do processo de maneira fácil, rápida e inteligente.

A ferramenta contém recursos de planejamento de *releases*, *sprints*, construção de *roadmaps*, rastreamento de requisitos, etc. Uma ferramenta completa com recursos adequados para a utilização neste projeto. Contudo, a aprendizagem da utilização concisa da ferramenta poderia ficar comprometida, tendo em vista o tempo destinado à realização do trabalho.

Segundo os idealizadores da ferramenta VersionOne, é possível perceber os seguintes pontos em relação à ferramenta:

- Alavancar um único sistema para o planejamento e acompanhamento de todos os épicos, histórias, temas, defeitos, tarefas, testes e questões. O VersionOne fornece uma visibilidade entre várias equipes, projetos e portfólios ágeis, proporcionando um ambiente centralizado, onde todos os *Stakeholders* - executivos, gerentes, proprietários do produto, desenvolvedores e testadores - podem facilmente trabalhar juntos, independentemente da localização;
- Construído a partir do zero para suportar os processos e fluxos de trabalho de desenvolvimento de *software* ágeis, o VersionOne acelera a adoção da metodologia ágil. Há orientação para os membros da equipe em cada etapa do processo de desenvolvimento ágil. O VersionOne reforça continuamente as melhores práticas ágeis, sem ser prescritivo;

- Há possibilidade de fornecer às equipes a flexibilidade para trabalhar da maneira que elas querem. Fluxo de trabalho e espaços de trabalho personalizados. O VersionOne permite fazer o caminho ágil e adaptar os processos à medida que evoluem. Ao invés de impor a todos um tamanho único de abordagem, é possível selecionar a partir de modelos previamente concebidos e personalizados (metodologia Scrum, XP, Kanban, AgileUP, etc).

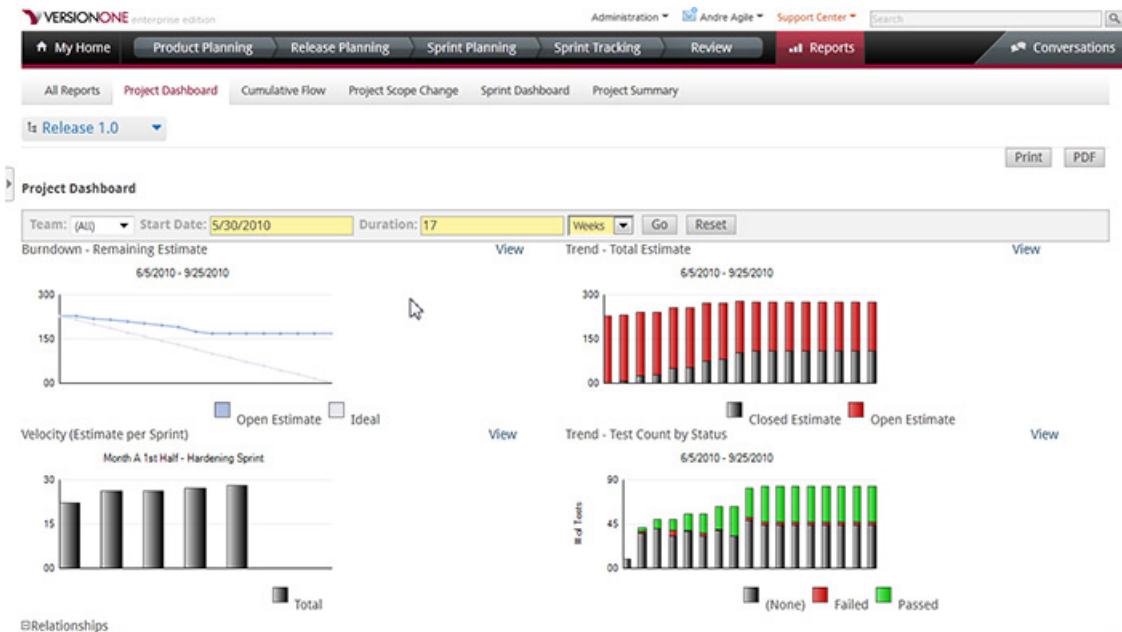


Figura 4 – Ferramenta VersionOne de Gestão de Requisitos.

8.1.2 A Ferramenta Rally *Software*

A ferramenta é fácil e agradável de utilizar. Ao realizar o cadastramento, informando o nome do projeto, bem como suas características, o grupo de Requisitos de *Software* conseguiu obter facilmente uma visão das funcionalidades. A plataforma Rally reúne gerenciamento ágil de projetos, juntamente com os gerenciamentos de requisitos e defeitos. Dessa maneira, a equipe poderá contemplar uma imagem em tempo real dos compromissos do projeto.

Como retratado anteriormente, o grupo da disciplina de Requisitos de *Software* adotou a metodologia ágil para trabalhar com requisitos. De acordo com o processo que foi desenhado, é possível contemplar as funcionalidades requeridas na ferramenta.

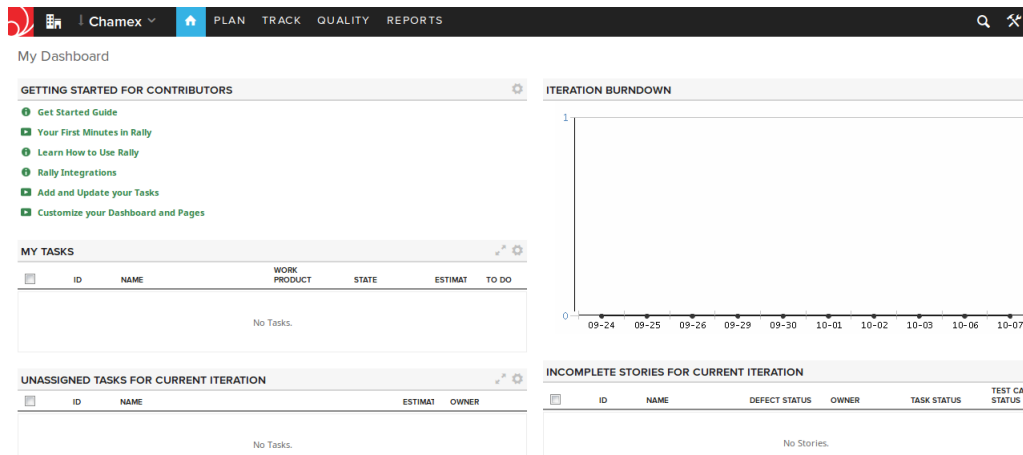


Figura 5 – Ferramenta Rally *Software* de Gestão de Requisitos.

8.1.3 Critérios Adotados para Avaliação

Para realizar uma escolha concisa de uma ferramenta para gerência de requisitos a ser utilizada neste trabalho, foi necessário estabelecer alguns critérios. Os critérios foram adotados segundo aspectos ergonômicos de *software* (CYBIS, 2011) e, adicionalmente, aspectos de suporte das ferramentas às atividades da Engenharia de Requisitos. Inicialmente, a partir de indicações de monitores da disciplina de Requisitos de *Software* e de leitura prévia de artigos, as ferramentas colocadas para investigação foram: VersionOne e Rally *Software*. Os tópicos abaixo exibem os critérios adotados para avaliação das ferramentas:

- **Multi-Plataforma:** Suporte da ferramenta em ser utilizada nos sistemas operacionais mais comuns (Windows, Linux e MacOS);
- **Web:** Possibilidade do sistema em ser utilizado via *web* através dos *browsers* mais comuns (Internet Explorer, Mozilla Firefox, Opera e Google Chrome);
- **Cadastro de *User Stories*:** Suporte do sistema para registro das *user stories*, bem como alteração das informações de data, prioridade, resumo, critérios de aceitação, dentre outros;
- **Gerência de iterações:** Permissão para registrar iterações e associá-las às *releases*, datas, pontuações de esforço, tempo gasto, dentre outros;
- **Visão do processo:** Capacidade da ferramenta em apresentar o fluxo de atividades;
- **Rastreabilidade:** Possibilidade da ferramenta em oferecer algum tipo de rastreabilidade, seja ela vertical ou horizontal;
- **Alocação de Recursos Humanos:** Permissão para alocar recursos humanos dentro das iterações de forma dinâmica;

- **Usabilidade:** Este critério avalia se a ferramenta é intuitiva, ou seja, de fácil utilização e não demanda alta quantidade de tempo para entender as suas funcionalidades e como utilizá-las;
- **Livre:** Ferramenta gratuita, sem ou com poucas limitações de recursos.

8.2 Escolha da Ferramenta de Gestão de Requisitos

As ferramentas apresentaram as mesmas funcionalidades, sendo que esse aspecto foi constatado em uma comparação. Contudo, a ferramenta *Rally Software* apresenta uma versão, denominada *Community*, que reúne os principais recursos avaliados e não há cobrança para utilização.

A ferramenta *VersionOne*, embora possua vários recursos e uma boa usabilidade, após o período de 30 (trinta) dias, alguns recursos seriam retirados, prejudicando o desenvolvimento do trabalho.

A Tabela 3 descreve a comparação entre as duas ferramentas segundo os critérios adotados.

Critérios	VersionOne	Rally <i>Software</i>
Multi-Plataforma	✓	✓
<i>Web</i>	✓	✓
Cadastro de <i>User Stories</i>	✓	✓
Gerência de Iterações	✓	✓
Visão do Processo	✓	✓
Rastreabilidade	✓	✓
Alocação de Recursos Humanos	✓	✓
Usabilidade	✓	✓
Livre	X	✓

Tabela 3 – Tabela de Comparação das Ferramentas.

9 Considerações Finais

A partir do desenvolvimento dessa primeira parte do trabalho proposto pela disciplina de Requisitos de *Software*, foi possível constatar que uma boa Engenharia de Requisitos é um passo essencial para o desenvolvimento de um bom produto, em qualquer ocasião.

É importante ressaltar que deve haver uma preocupação em levantar requisitos de alta qualidade. Adicionalmente, a partir da elaboração de procedimentos de controle em um processo de Engenharia de Requisitos, os requisitos atingem patamares cada vez maiores na escala de qualidade.

Referências

CYBIS, W. de A. *ErgoList - Menu de Checklists*. Universidade Federal de Santa Catarina, 2011. Disponível em: <<http://www.labiutil.inf.ufsc.br/ergolist/check.htm>>. Citado na página 55.

FILHO, W. de P. P. *Engenharia de Software: fundamentos, métodos e padrões*. [S.l.]: LTC, 2009. Citado na página 7.

LEFFINGWELL, D. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise (Agile Software Development Series)*. 1. ed. [S.l.]: Addison-Wesley Professional, 2011. Citado 5 vezes nas páginas 12, 15, 16, 20 e 25.

PCMAGAZINE. *PC Magazine Encyclopedia - Meaning of Prototyping*. The Computer Language Company Inc., 2014. Disponível em: <<http://www.pcmag.com/encyclopedia/term/49886/prototyping>>. Citado na página 30.

SCHWABER, K. *Guia do Scrum*. Scrum Alliance, 2009. Disponível em: <http://www.trainning.com.br/download/GUIA_DO_SCRUM.pdf>. Citado na página 32.