



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

Processo de Engenharia de Requisitos - Matriz Engenharia de Energia

Autor: Edson Gomes, Emilie Moraes, Filipe Ribeiro, Hugo
Martins

Orientador: George Marsicano Côrrea, MSc

Brasília, DF
5 de Maio de 2015



Edson Gomes, Emilie Moraes, Filipe Ribeiro, Hugo Martins

Processo de Engenharia de Requisitos - Matriz Engenharia de Energia

Relatório 1 da disciplina de Requisitos
de Software submetido na Faculdade UnB
Gama da Universidade de Brasília.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: George Marsicano Côrrea, MSc

Brasília, DF

5 de Maio de 2015

Lista de ilustrações

Figura 1 – Processo de Engenharia de Requisitos	12
Figura 2 – Subprocesso Executar <i>Release</i>	17
Figura 3 – Supprocesso Executar Iteração	19
Figura 4 – Atividades do Processo com a Gerência de Requisitos do CMMI	26
Figura 5 – Atividades do Processo com o Desenvolvimento de Requisitos do CMMI	27
Figura 6 – Estratégia de rastreabilidade	33
Figura 7 – Primeira Parte do Cronograma	35
Figura 8 – Segunda Parte do Cronograma	36
Figura 9 – Terceira Parte do Cronograma	36
Figura 10 – Quarta Parte do Cronograma	37
Figura 11 – Quinta Parte do Cronograma	37
Figura 12 – Sexta Parte do Cronograma	38
Figura 13 – Tela inicial da ferramenta Tiger-Pro	41
Figura 14 – Tela inicial da ferramenta TraceCloud	42
Figura 15 – Tela inicial da ferramenta Caliber	43
Figura 16 – Quadro comparativo entre ferramentas.	45

Sumário

1	INTRODUÇÃO	5
1.1	Organização do Trabalho	5
2	CONTEXTO DA EMPRESA	7
2.1	O Problema	7
2.2	Possível Solução	7
3	JUSTIFICATIVA	9
4	PROCESSO DE ENGENHARIA DE REQUISITOS	11
4.1	Atividades	13
4.1.1	Estabelecer Tema de Investimento	13
4.1.2	Levantar Épicos	13
4.1.3	Fazer reunião de validação dos épicos	14
4.1.4	Levantar <i>Features</i>	14
4.1.5	Fazer reunião de validação das <i>features</i>	15
4.1.6	Identificar Requisitos Não Funcionais	15
4.1.7	Construir Visão	16
4.1.8	Construir <i>Roadmap</i>	17
4.1.9	Priorizar <i>features</i>	18
4.1.10	Escrever histórias	18
4.1.11	Realizar reunião de planejamento da Iteração	19
4.1.12	Especificar histórias	20
4.1.13	Desenvolver histórias	20
4.1.14	Fazer reunião de revisão e retrospectiva da Iteração	21
4.2	Artefatos	21
4.3	Papéis	22
5	MODELO DE MATURIDADE NO PROCESSO	25
5.1	Gerência de Requisitos	25
5.2	Desenvolvimento de Requisitos	27
6	ELICITAÇÃO	29
6.1	Técnicas de Elicitação de Requisitos	29
7	TÓPICOS DE GERENCIAMENTO DE REQUISITOS	31
7.1	Atributos de Requisitos	31

7.2	Rastreabilidade de Requisitos	32
8	PLANEJAMENTO	35
9	FERRAMENTAS DE GERENCIAMENTO DE REQUISITOS	39
9.1	Ferramentas encontradas	39
9.1.1	Tiger Pro	39
9.1.2	TraceCloud	39
9.1.3	Caliber	40
9.2	Análise das Ferramentas	40
9.2.1	Tiger Pro	40
9.2.2	TraceCloud	41
9.2.3	Caliber	43
9.3	CrITÉrios de Avaliação das Ferramentas	43
9.4	Escolha da Ferramenta	45
10	CONSIDERAÇÕES FINAIS	47
	Referências	49

1 Introdução

Segundo [Pressman \(2011\)](#), Engenharia de Requisitos é “um amplo espectro de tarefas e técnicas que levam a um entendimento dos requisitos”. [Jitnah, Han e Steele \(1995\)](#) afirmam que é o processo através do qual as necessidades dos usuários são identificadas e expressas em um artefato.

A partir disso, é necessário estudar o contexto de negócio, definir técnicas de elicitação, definir estratégias de gerenciamento de requisitos, definir atividades, artefatos, papéis do processo e definir uma ferramenta que dê suporte ao processo e gerenciamento definidos.

Dessa forma, este trabalho tem como objetivo definir um processo de Engenharia de Requisitos para uma empresa júnior, chamada Matriz Engenharia de Energia, com o intuito de solucionar um problema da organização por meio de um sistema de software, realizando todas as atividades supracitadas.

Com o estudo do contexto de negócio e através da análise de perfil do cliente e da equipe, o processo estabelecido neste trabalho possui uma abordagem adaptativa baseada nos *frameworks* SAFe e Scrum.

1.1 Organização do Trabalho

Este relatório está organizado nos seguintes capítulos: Contexto de Negócio, Justificativa da Abordagem, Processo de Engenharia de Requisitos, Elicitação, Tópico de Gerenciamento de Requisitos, Planejamento, Ferramentas de Gestão de Requisitos e Considerações Finais.

No *Capítulo 2: Contexto de Negócio* é apresentado o contexto no qual o processo definido irá atuar.

No *Capítulo 3: Justificativa da Abordagem* são apresentados argumentos que implicaram na escolha da abordagem adaptativa para este trabalho.

No *Capítulo 4: Processo de Engenharia de Requisitos* é descrito o processo de Engenharia de Requisitos definido com suas atividades, tarefas, papéis e artefatos.

No *Capítulo 5: Elicitação* são apresentadas as técnicas de elicitação a serem utilizadas na execução do processo definido.

No *Capítulo 6: Tópico de Gerenciamento de Requisitos* são abordados os atributos de requisitos e a estratégia de rastreabilidade, para gerenciamento dos requisitos.

No *Capítulo 7: Planejamento* é apresentado um cronograma de atividades definido para guiar o andamento das atividades.

No *Capítulo 8: Ferramentas de Gerenciamento de Requisitos* são apresentadas algumas ferramentas de gerenciamento de requisitos selecionadas para avaliação. A avaliação e a escolha de uma ferramenta também é abordada nesse capítulo.

No *Capítulo 9: Considerações Finais* é apresentada a conclusão obtida com este relatório e os resultados esperados com o próximo trabalho.

2 Contexto da Empresa

O contexto proposto, se refere a empresa júnior Matriz - Engenharia de Energia, que é formada por alunos da Engenharia de Energia da Universidade de Brasília, Campus Gama. O objetivo principal da Matriz é a iniciação dos alunos do curso no mercado de trabalho fornecendo aos clientes soluções inteligentes referentes a energia e sustentabilidade.

A empresa visa atender a demanda de residências, condomínios, empresas de médio e pequeno porte. A Matriz, tenta desenvolver serviços exclusivos como estudos de viabilidade energética, projetos de geração distribuída de energia, adequação tarifária, controle e gestão inteligente de energia, correção de fator de potência, auxílio e suporte em comercialização de energia, desenvolvimento de projetos elétricos além do oferecimento de cursos e treinamentos voltados à capacitação.

2.1 O Problema

Como já dito, a Matriz vêm com a proposta de ser uma Empresa Júnior que proporcione para os alunos de graduação uma experiência empresarial. Por ser nova, a empresa não possui um processo bem definido que resulte na aquisição de novos clientes. Inicialmente o que foi definido, consistia em investir no uso de redes sociais para difundir suas ideias, no estudo de mercado no que tange empresas que mais gastam energia e enviar via telefone propostas dos serviços que a empresa disponibiliza. Feito isso, optou-se então pelo uso de cartazes nos corredores da Faculdade do Gama (FGA), porém nenhuma das estratégias seguidas surtiu efeito no que diz respeito a adesão de sócios e clientes.

2.2 Possível Solução

A partir deste quadro, o grupo da disciplina de Requisitos de Software, junto com um grupo da disciplina de Modelagem de Processos, foi encubido de criar estratégias criativas para obter uma maior adesão a Matriz. O problema será solucionado usando conceitos referentes a Engenharia de Requisitos, juntos a metodologias de desenvolvimento adaptativas, desenhando assim um processo que consistirá desde o nível de análise do que de fato se necessita para reverter esse quadro, até o ponto de apresentar uma solução em software para o problema proposto.

Serão realizadas portanto, interações com os interessados no produto final para que os mesmos possam fornecer dados que possibilitem além de um auxílio aos responsáveis

por desenvolver o projeto, negociações e documentações de possíveis planos que serão mudados.

3 Justificativa da Abordagem

A abordagem definida foi a adaptativa com o uso dos *frameworks Scaled Agile Framework* (SAFe) e Scrum. Para a definição da abordagem a ser utilizada, três aspectos foram analisados: o contexto de negócio no qual este trabalho está inserido, a equipe e a interação entre a equipe e o cliente.

No contexto de negócio foram analisadas algumas características como: tamanho da empresa, estabilidade da empresa e demanda por maior formalização.

A empresa Matriz - Engenharia de Energia por se tratar de uma empresa júnior consiste em uma empresa pequena, que possui poucas pessoas envolvidas e não demanda grande formalização.

Segundo [Alves e Alves \(2008\)](#), “as metodologias tradicionais têm uma orientação para que as atividades e artefatos sejam formalmente documentados e controlados. Por outro lado, as metodologias ágeis têm seu foco na iteratividade dos interessados no projeto.” Dessa forma, como a empresa não demanda muitas formalizações, a abordagem adaptativa, ou ágil, se adequa a esse perfil.

No aspecto de equipe, características como tamanho da equipe, experiências anteriores e interação semanal foram levadas em consideração. A equipe, formada por quatro integrantes, possui pouca experiência em Engenharia de Requisitos tanto na abordagem tradicional, quanto na adaptativa. O que poderia levar a escolha da abordagem tradicional. As interações entre os membros da equipe foram definidas em quatro vezes na semana. [Alves e Alves \(2008\)](#) relatam que os valores ágeis estão alinhados com dinâmicas de pequenas e médias organizações.

Para o aspecto de interação com o cliente foi levada em consideração a quantidade de reuniões predefinidas, que consiste em uma reunião semanal. Desse modo, também analisando a proximidade com o cliente.

Os processos ágeis são caracterizados por [Pressman \(2011\)](#) como processos com capacidade de adaptação e que sejam incrementais, tendo em vista as mudanças e o entendimento dos requisitos, e a participação direta do cliente através do seu constante *feedback*. A interação semanal estabelecida com o cliente permite esse constante *feedback*.

[Leffingwell \(2011\)](#) afirma que a metodologia ágil se concentra em dois aspectos principais: melhor compreensão das necessidades e construção de software de qualidade.

Fazer essa afirmação não significa dizer que as metodologias tradicionais não possuem essas características. O processo unificado, tem algumas características semelhantes com a metodologia adaptativa, como: ser baseado em interações e estabelecer interação

com o cliente ,mas mesmo assim, nesse tipo de abordagem a concentração acaba sendo no processo. ((LEFFINGWELL, 2011) ; (VASCO; VITHOFT; ESTANTE, 2006))

A escolha do *framework* SAFe foi realizada devido ao seu processo de Engenharia de Requisitos (ER), pois o SAFe engloba as atividades fundamentais da ER. E de acordo com Leffingwell (2011), foi um modelo criado para suportar a necessidade de *feedback* para o negócio. Todavia o SAFe será utilizado apenas como um *framework* estrutural que definirá os artefatos e papéis e o Scrum guiará a execução do trabalho através dos marcos de tempo, de validação e os eventos, como: Reunião de Planejamento da *Sprint* e Reunião de Retrospectiva da *Sprint*, como proposto por Shwaber e Sutherland (2013).

O processo unificado embora se adeque a alguns aspectos analisados, como analisado por Vasco, Vithoft e Estante (2006), essa abordagem é melhor aplicável em grandes projetos e tem tarefas e papéis muito bem definidos. Já o SAFe e o Scrum permitem uma maior dinâmica na equipe.

4 Processo de Engenharia de Requisitos

O processo de Engenharia de Requisitos foi estabelecido com base no SAFe combinado com o Scrum.

O SAFe é dividido em três níveis: Portfólio, Programa e Time. Em cada nível são representados os requisitos em diferentes graus de abstração, como: Temas de Investimento, Épicos, *Features* e Histórias. Cada nível também possui papéis responsáveis por criar e manter esses requisitos, como: *Product Portfolio Manager*, *Product Manager* e *Product Owner*. O SAFe é baseado no Scrum e possui características, alguns artefatos e papéis semelhantes. (LEFFINGWELL, 2011)

O Scrum é formado por eventos, papéis e artefatos. Os eventos são: *Sprint*, Reunião de Planejamento da *Sprint*, Reunião diária e Reunião de Retrospectiva da *Sprint*. Os papéis são: *Product Owner*, *Scrum Master* e Time de Desenvolvimento. Os artefatos são: *Backlog* do Produto, *Backlog da Sprint* e Incremento. (SHWABER; SUTHERLAND, 2013).

Dessa forma, foi estabelecido um processo, que pode ser visto na Figura 1, adequado ao contexto do trabalho. É composto das seguintes atividades: Estabelecer Tema de Investimento, Levantar épicos, Fazer reunião de validação dos épicos, Levantar *Features*, Fazer reunião de validação das *Features*, Construir Visão, Construir *Roadmap*, Identificar Requisitos Não Funcionais e dos subprocesso Executar *Release*, tem Priorizar *Features* e Escrever Histórias e no subprocesso Executar Iteração tem Realizar reunião de Planejamento da Iteração, Especificar Histórias, Desenvolver Histórias e Realizar reunião de revisão e Retrospectiva da Iteração.

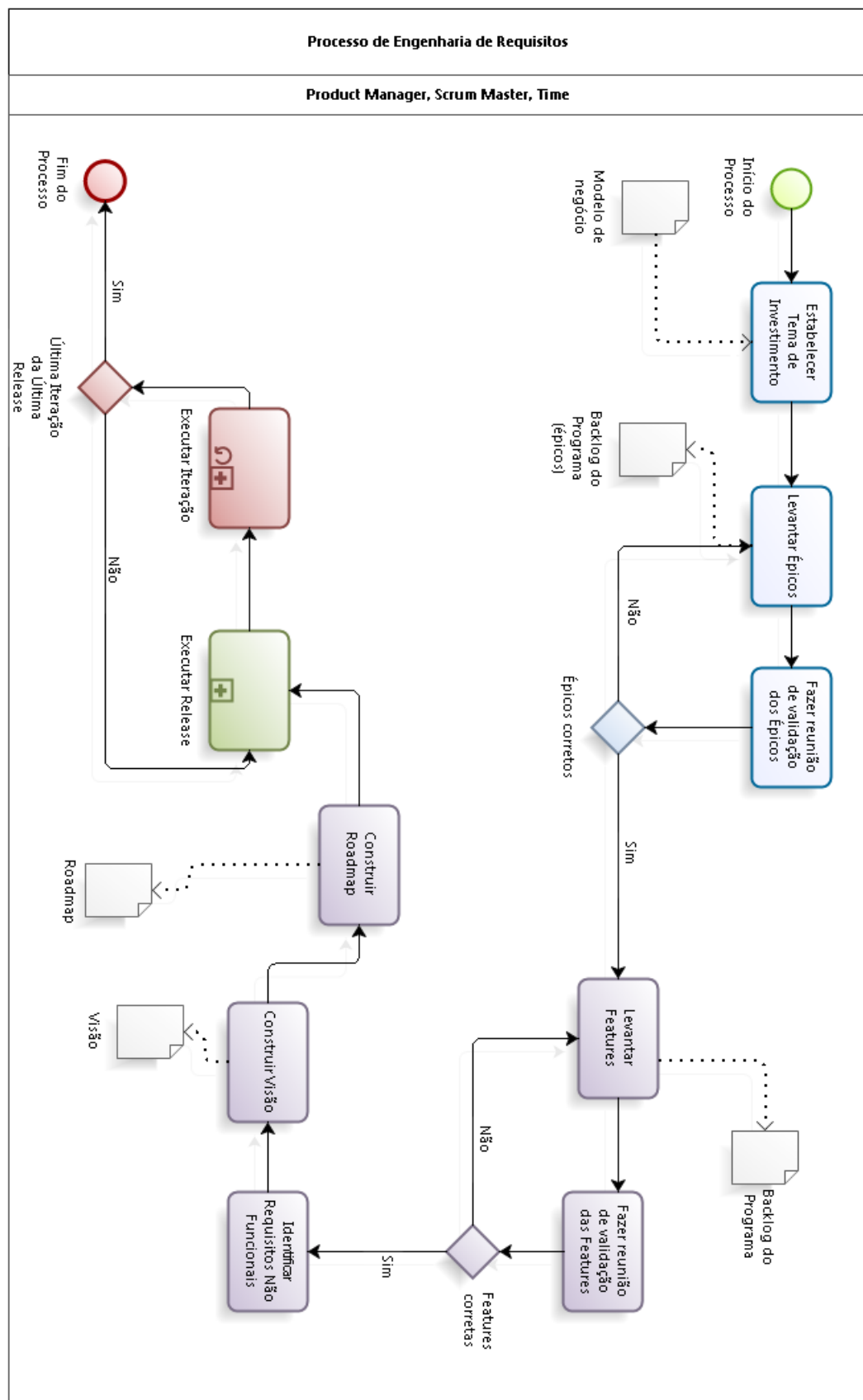


Figura 1 – Processo de Engenharia de Requisitos

4.1 Atividades

4.1.1 Estabelecer Tema de Investimento

Descrição: Essa atividade consiste na definição do tema de investimento a partir do *Workshop* de Requisitos realizado.

Tarefas:

- *Fazer Análise Documental:* Analisar documentos da empresa e o modelo de negócios produzido pela equipe de Modelagem.
- *Fazer Workshop de Requisitos:* Reunião com o cliente e o time para levantar requisitos de mais alto nível. Consiste em uma técnica de elicitação definida no Capítulo 6.1.
- *Validar Tema de Investimento:* Escrever o tema de investimento e confirmar com o cliente se o Tema de Investimento estabelecido está correto.

Participantes: *Product Manager, Scrum Master, Time*

Entradas: Modelo de negócio

Saídas: Anotações do *Workshop*

4.1.2 Levantar Épicos

Descrição: Essa atividade consiste no levantamento dos épicos com o *Product Manager* através do *Workshop* de Requisitos.

Tarefas:

- *Fazer Workshop de Requisitos:* Reunião com o cliente e o time para levantar requisitos de mais alto nível. Consiste em uma técnica de elicitação definida no Capítulo 6.1.
- *Escrever Épicos:* A partir das anotações feitas no *Workshop*, escrever os épicos no *Backlog* do Programa.
- *Manter Rastreabilidade:* Registrar na ferramenta cada épico como derivado do tema de investimento.

Participantes: *Product Manager, Scrum Master, Time*

Entradas: Anotações do *Workshop*

Saídas: *Backlog* do Programa

4.1.3 Fazer reunião de validação dos épicos

Descrição: Essa atividade consiste na apresentação dos épicos especificados para o *Product Manager*, de modo a validar se os épicos especificados estão corretos e correspondem ao esperado.

Tarefas:

- *Validar Épicos:* Confirmar em reunião com o cliente se os épicos estabelecidos estão corretos.
- *Escrever Épicos:* Se houver alguma mudança solicitada pelo cliente, escrever os épicos refinados no *Backlog* do Programa.

Participantes: *Product Manager, Scrum Master, Time*

Entradas: *Backlog* do Programa

Saídas: Não se aplica

4.1.4 Levantar *Features*

Descrição: Essa atividade consiste em listar as *Features*, a partir dos épicos, que são as tarefas ou os “serviços” que o sistema deve fornecer para atender as necessidades das partes interessadas. Deve-se observar se as *features* condizem com ou traduzem de forma clara os Épicos definidos previamente, e se através delas é possível escrever as Histórias de Usuário.

Tarefas:

- *Listagem de Features*: Listar *Features* a partir dos Épicos de Portfólio ou a partir de outras *Features*.
- *Adição e Edição de Features*: Adicionar novas *Features* ou modificar as *Features* já levantadas de acordo com as mudanças sugeridas na Reunião de Validação.
- *Manter Rastreabilidade*: Registrar na ferramenta de qual épico é cada *Feature*.

Participantes: *Product Manager, Scrum Master, Time*

Entradas: *Backlog* do Programa (épicos)

Saídas: *Backlog* do Programa

4.1.5 Fazer reunião de validação das *features*

Descrição: Essa atividade consiste na apresentação das *features* especificadas para o *Product Manager*, de modo a validar se estão corretos e correspondem ao esperado.

Tarefas:

- *Listagem de Features*: Listar as *Features* elicítadas e detalhadas até o momento;
- *Comparação Features-Épicos*: Comparar as *Features* e os Épicos, validando se as *Features* expressam as iniciativas contidas nos Épicos;
- *Corrigir Possíveis Falhas*: Identificar equívocos no levantamento ou elicitação das *Features* e propor mudanças.

Participantes: *Product Manager, Scrum Master, Time*

Entradas: *Backlog* do Programa

Saídas: Não se aplica

4.1.6 Identificar Requisitos Não Funcionais

Descrição: Nesta atividade são identificados e descritos os requisitos não funcionais do sistema e são armazenados no *Backlog* do Programa.

Tarefas:

- *Identificação de Requisitos não Funcionais*: Identificar e descrever os requisitos não funcionais
- *Armazenamento no Backlog*: Armazenar no *Backlog* do Programa

Participantes: *Product Manager, Scrum Master, Time*

Entradas: Não se aplica

Saídas: *Backlog* do Programa

4.1.7 Construir Visão

Descrição: Nesta atividade, deve-se compilar as *Features*, com os requisitos não funcionais, incluindo elementos regulatórios ou outros padrões de conformidade, e qualquer restrição de design. A partir disso, é possível descrever um panorama da solução a ser desenvolvida, refletindo as necessidades das partes interessadas e os recursos propostos para atender essas necessidades.

Tarefas:

- *Reunir Artefatos*: Reunir *Features*, Requisitos não Funcionais, restrições e padrões.
- *Sintetizar e Integrar*: Sintetizar todas as entradas, integrando-as em uma visão holística (global) e coesa do projeto.
- *Planejamento do RoadMap*: Planejar o *RoadMap* de entregas das *Features*.

Participantes: *Product Manager, Scrum Master, Time*

Entradas: *Backlog* do Programa, Requisitos Não funcionais

Saídas: Visão

4.1.8 Construir *Roadmap*

Descrição: A partir da Visão, deve-se elaborar uma espécie de roteiro, que situa e comunica a equipe e o programa em relação ao alinhamento dos objetivos de negócios, e fornece visibilidade das entregas ao longo de um cronograma de curto prazo. Esse roteiro divide as *features* nas *Releases*.

Tarefas:

- *Reunir Documento de Visão:* Reunir o documento de visão
- *Definir Prioridades:* Priorizar *Features* no *Backlog* do programa.
- *Alocamento de Features:* Alocar *Features* em *Releases*.
- *Escrever:* Escrever histórias

Participantes: *Product Manager, Scrum Master, Time*

Entradas: Visão

Saídas: *Roadmap*

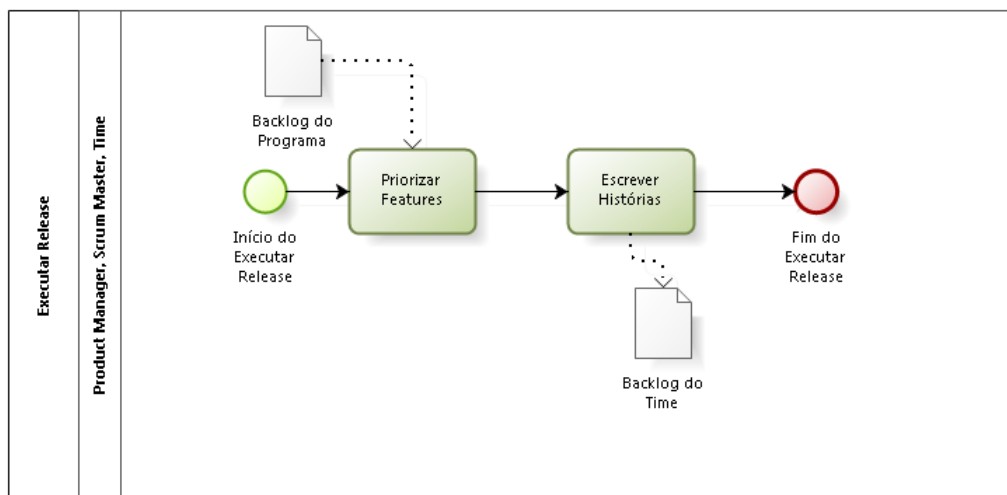


Figura 2 – Subprocesso Executar *Release*

A partir do Subprocesso da Figura 2 o processo contém as seguintes atividades.

4.1.9 Priorizar *features*

Descrição: Essa atividade consiste na priorização das *features* para a *Release*.

Tarefas:

- *Reunir Features*: Reunir *features* envolvidas na *Release*;
- *Estudar as Features*: Estudar as *features* para melhor entendimento de cada uma;
- *Definir Indicador de Complexidade*: Definir um identificador para saber quais demandam mais trabalho (horas de serviço);
- *Ordenar*: A partir dos resultados obtidos via análise, ordenar das que mostraram maiores métricas de esforço, até as menores, de modo que as mais difíceis sejam executadas o quanto antes.

Participantes: *Product Manager*, *Scrum Master*, Time

Entradas: *Backlog* do Programa

Saídas: Não se aplica

4.1.10 Escrever histórias

Descrição: Essa atividade consiste na escrita das histórias em um nível macro, derivadas das *features*, para composição do *Backlog* do Time.

Tarefas:

- *Identificar*: Identificar *features*;
- *Analisar*: Analisar o nível de prioridade das *features*;
- *Alocar*: Alocar *features* nas iteração;
- *Instruir*: Auxílio do Scrum Master ao Product Manager a fim de ensinamento; de técnicas de escritas de histórias de usuário;
- *Escrita*: Escrever histórias;
- *Armazenamento*: Armazenar histórias no *Backlog* do time.

- *Manter Rastreabilidade*: Registrar na ferramenta de qual *Feature* é cada história.

Participantes: *Product Manager, Scrum Master, Time*

Entradas: *Backlog do Programa*

Saídas: *Backlog do Time*

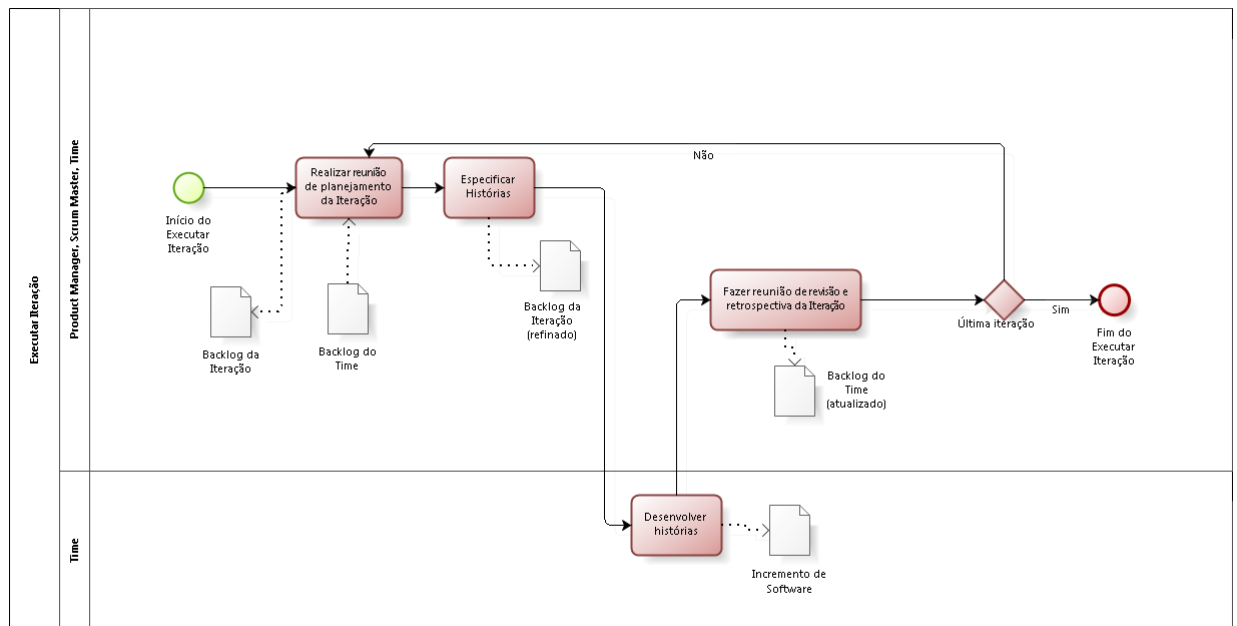


Figura 3 – Supprocesso Executar Iteração

A partir do Subprocesso da Figura 3 o processo contém as seguintes atividades.

4.1.11 Realizar reunião de planejamento da Iteração

Descrição: Essa atividade consiste no planejamento da iteração através da alocação das histórias definidas no *Backlog* do Time para o *Backlog* da iteração.

Tarefas:

- *Alocação das Histórias*: Alocação das histórias no *Backlog* da iteração
- *Armazenamento de Artefatos*: Armazenar no *Backlog* de Iteração

Participantes: *Product Manager, Scrum Master, Time*

Entradas: *Backlog* do Time

Saídas: *Backlog* da Iteração

4.1.12 Especificar histórias

Descrição: Essa atividade consiste na escrita mais detalhada das histórias que irão compor a iteração.

Tarefas:

- *Detalhamento das Histórias:* Detalhar histórias da iteração
- *Testes de aceitação:* Escrever testes de aceitação das histórias da iteração
- *Armazenamento de Artefatos:* Armazenar no *Backlog* de Iteração
- *Atualizar:* Atualizar *Backlog* do Time

Participantes: *Product Manager, Scrum Master, Time*

Entradas: *Backlog* da Iteração

Saídas: *Backlog* da Iteração (refinado)

4.1.13 Desenvolver histórias

Descrição: Essa atividade consiste no desenvolvimento e teste das histórias contidas no *Backlog* da Iteração. Tem uma duração de 1 semana.

Tarefas:

- *Desenvolver Histórias:* Implementar as histórias contidas no *Backlog* da Iteração.
- *Testar Histórias:* Realizar testes unitários do código implementado na iteração e testes funcionais com base nos testes de aceitação.

Participantes: Time

Entradas: *Backlog* da Iteração

Saídas: Incremento de Software

4.1.14 Fazer reunião de revisão e retrospectiva da Iteração

Descrição: Essa atividade consiste na validação das histórias implementadas através da execução dos testes de aceitação.

Tarefas:

- *Analise de Artefatos:* Analisar artefatos gerados na iteração;
- *Avaliar Pontos Positivos:* Definir pontos que ocorreram bem;
- *Definir Pontos de Melhoria:* Definir pontos que necessitam de melhoria;
- *Definir Novas Metas:* Definir novas metas de melhoria a serem alcançadas na nova iteração.
- *Atualizar o Backlog:* Atualizar o *Backlog* do Time.

Participantes: *Product Manager*, Time

Entradas: Incremento de Software

Saídas: *Backlog* da Iteração (atualizado)

4.2 Artefatos

- *Backlog* do Programa

O *Backlog* do Programa é responsável por manter os épicos e as *features*. É criado na atividade Levantar Épicos e pode ser atualizado nas atividades: Fazer reunião de validação dos épicos, Levantar *Features*, Especificar *Features*, Fazer reunião de validação das *Features*, Construir Visão, Construir *Roadmap*, Identificar Requisitos Não Funcionais e durante as atividades do subprocesso Executar *Release*.

- Visão

O Visão consiste em um documento que contém os requisitos funcionais, requisitos não funcionais, incluindo elementos regulatórios ou outros padrões de conformidade, e qualquer restrição de design. Também contém um panorama da solução a ser desenvolvida, refletindo as necessidades das partes interessadas e os recursos propostos para atender essas necessidades. É gerado na atividade Construir Visão.

- *Roadmap*

O *Roadmap* consiste na alocação das *features* em *Releases*, através da determinação de datas e prioridades. O *Roadmap* é definido na atividade: Construir *Roadmap*.

- *Backlog* do Time

O *Backlog* do Time é responsável por manter as histórias levantadas a partir das *features*. É criado na atividade Escrever histórias da primeira *Release* e pode ser atualizado nas atividades: Escrever histórias, Especificar histórias, Realizar reunião de planejamento da Iteração, Fazer reunião de revisão e retrospectiva da Iteração.

- *Backlog* da Iteração

O *Backlog* da Iteração é responsável por manter as histórias da iteração atual. É criado na atividade Realizar reunião de planejamento da Iteração e pode ser atualizado nas atividades: Especificar histórias.

- Incremento de Software

O Incremento de Software consiste na solução gerada a cada Iteração. É gerada na atividade de Desenvolver Histórias e validada na atividade de Reunião de Revisão e retrospectiva da Iteração.

4.3 Papéis

- *Product Manager* (PM)

O *Product Manager* consiste na equipe de Modelagem de Processos e suas responsabilidades são:

-> Escrever as histórias;

-> Gerenciar o escopo;

-> Definir o visão;

-> Criar o *Roadmap*.

- *Scrum Master*

O *Scrum Master* é um membro da equipe de Requisitos responsável por:

-> Verificar se o processo está sendo realizado conforme planejado;

-> Auxiliar o PM nas suas atividades;

-> Servir de elo de ligação entre o PM e o Time.

- Time

O Time consiste na equipe de Requisitos e é responsável por:

-> Escrever os testes de aceitação das histórias;

-> Desenvolver as histórias.

5 Modelo de maturidade no processo

Modelo de maturidade consiste em uma série de indicadores de qualidades de processo, fornecendo tal indicação em nível de maturidade. (PRESSMAN, 2011). Para Pressman (2011), “sua finalidade é proporcionar uma indicação geral da “maturidade do processo” exibida por uma organização de software”.

Para as avaliações de processos de software existem dois modelos CMMI e o MPS.BR. “O CMMI ® (Capability Maturity Model ® Integration – Modelo Integrado de Maturidade e de Capacidade) é um modelo de maturidade para melhoria de processo, destinado ao desenvolvimento de produtos e serviços, e composto pelas melhores práticas associadas a atividades de desenvolvimento e de manutenção que cobrem o ciclo de vida do produto desde a concepção até a entrega e manutenção.” (CARNEGIE MELLON UNIVERSITY, 2010)

O MPS.BR é um programa que visa a Melhoria de Processos de Software e Serviços em duas metas: meta técnica e meta de negócio. A meta técnica está relacionada com a melhoria do modelo MPS, e meta de negócio consiste na difusão do modelo para auxiliar pequenas, médias (foco) e grandes organizações. (SOFTEX, 2012)

Para esse trabalho foi estabelecido o uso do CMMI, por possuir suas práticas mais detalhadas. Assim, nesse capítulo é apresentado um mapeamento das práticas do CMMI que estão relacionadas com as atividades do processo de Engenharia de Requisitos definido. Essas práticas são provenientes do nível 2 do CMMI com o processo de Gerência de Requisitos e do nível 3 do CMMI com o processo de Desenvolvimento de Requisitos. Esse mapeamento pode ser visto nas Figuras 4 e 5.

5.1 Gerência de Requisitos

Os objetivos das práticas escritas nessa seção foram baseadas nas descrições contidas no CMMI for Development publicado pela Carnegie Mellon University (2010).

Na prática específica de **Entender os Requisitos** estão envolvidas as atividades que permitem um entendimento dos requisitos através das técnicas de elicitação definidas para a familiarização com o processo, tais como: Estabelecer tema de investimento, Levantar épicos, Levantar *Features*. Além disso, o entendimento dos requisitos pode ser obtido através de reuniões de validação com o cliente, atividades essas nomeadas como: Fazer reunião de validação dos épicos e fazer reunião de validação das *features*.

Na prática específica de **Obter Compromisso para os requisitos**, a medida que os requisitos evoluem, esta prática garante que ao longo do projeto seja obtido um

Gerência de Requisitos	
CMMI	ATIVIDADE DO PROCESSO
SP 1.1 Obter Entendimento dos Requisitos	<ul style="list-style-type: none"> - Estabelecer temas de investimento - Levantar épicos - Fazer reunião de validação dos épicos - Levantar Features - Fazer reunião para validação das Features
SP 1.2 Obter Comprometimento com os Requisitos	<ul style="list-style-type: none"> - Estabelecer temas de investimento - Levantar épicos - Fazer reunião de validação dos épicos - Levantar Features - Fazer reunião de validação das Features - Identificar requisitos não funcionais - Construir visão - Construir Roadmap - Priorizar Features - Escrever histórias - Realizar reunião de revisão e retrospectiva da Iteração - Especificar histórias - Desenvolver histórias
SP 1.3 Gerenciar Mudanças nos Requisitos	<ul style="list-style-type: none"> - Fazer reunião de validação dos épicos - Fazer reunião de validação das Features - Realizar reunião de revisão e retrospectiva da Iteração
SP 1.4 Manter Rastreabilidade Bidirecional dos Requisitos	<ul style="list-style-type: none"> - Estabelecer temas de investimento - Levantar épicos - Levantar Features
SP 1.5 Identificar inconsistências entre o projeto e os requisitos	<ul style="list-style-type: none"> - Fazer reunião de validação dos épicos - Fazer reunião de validação das Features - Realizar reunião de revisão e retrospectiva da Iteração

Figura 4 – Atividades do Processo com a Gerência de Requisitos do CMMI

comprometimento para com os requisitos em todas as suas fases. No processo definido neste trabalho desde a primeira atividade que envolvem os requisitos até aquelas que tratam deles em suas últimas fases, os requisitos têm a sua devida atenção para que o projeto seja desenvolvido com qualidade.

Durante o processo definido neste trabalho, os requisitos podem sofrer alterações por diversos motivos. Nesta prática específica, **Gerenciar mudanças de requisitos**, a medida que há mudanças no projeto, estas devem ser documentadas e devem ficar claras para todos os envolvidos para que não ocorram inconsistências ao decorrer do projeto. Por isso, a prática de gerenciar mudanças nos requisitos envolvem as seguintes atividades de validação: Fazer reunião de validação dos épicos, Fazer reunião de validação das *features* e Realizar reunião de revisão e retrospectiva de Iteração), pois são nelas que obtém-se validação das atividades realizadas e, caso ocorram mudanças, serão nelas que estas mudanças serão documentadas e esclarecidas.

Quando há um bom gerenciamento dos requisitos, a sua rastreabilidade dos requisitos é gerenciada desde a sua origem até a sua última fase de detalhamento. Na prática específica de **Manter a rastreabilidade bidirecional dos requisitos**, tarefas específicas das atividades, tais como relacionar épicos ao tema de investimento, relacionar

features aos épicos e relacionar histórias às *features*, esta rastreabilidade será garantida para que seja obtida a origem dos requisitos.

Na prática específica de **Identificar inconsistências entre o projeto e os requisitos**, deve-se procurar inconsistências no projeto em todos os seus processos e corrigí-las. Esta prática se encontra nas atividades de validação dos requisitos tais como, Fazer reunião de validação dos épicos, Fazer reunião de validação das *features* e Realizar reunião de revisão e retrospectiva da iteração, pois são nas validações que essas inconsistências serão identificadas e corrigidas.

5.2 Desenvolvimento de Requisitos

Os objetivos das práticas escritas nessa seção foram baseadas nas descrições contidas no CMMI for Development publicado pela [Carnegie Mellon University \(2010\)](#).

Desenvolvimento de Requisitos	
CMMI	ATIVIDADE DO PROCESSO
SP 1.1 Levantar Necessidades	<ul style="list-style-type: none"> - Estabelecer Tema de Investimento - Levantar Épicos - Levantar Features - Identificar Requisitos Não Funcionais - Escrever Histórias
SP 1.2 Desenvolver Requisitos do Cliente	<ul style="list-style-type: none"> - Estabelecer Tema de Investimento - Levantar Épicos - Levantar Features - Escrever Histórias - Especificar Histórias
SP 2.1 Estabelecer requisitos do produtos e dos componentes do produto	<ul style="list-style-type: none"> - Identificar Requisitos Não Funcionais
SP 3.2 Estabelecer uma definição de funcionalidades requerida	<ul style="list-style-type: none"> - Especificar Histórias
SP 3.3 Analisar Requisitos	<ul style="list-style-type: none"> - Estabelecer Tema de Investimento - Levantar Épicos - Fazer reunião de validação dos Épicos - Levantar Features - Fazer reunião de validação das Features - Construir Visão
SP 3.5 Validar Requisitos	<ul style="list-style-type: none"> - Fazer reunião de validação dos épicos - Fazer reunião de validação das Features - Fazer reunião de revisão e retrospectiva da Iteração

Figura 5 – Atividades do Processo com o Desenvolvimento de Requisitos do CMMI

Na prática específica **Levantar necessidades** as expectativas e necessidades dos usuários são identificadas. Nas atividades Estabelecer Tema de Investimento, Levantar Épicos, Levantar *Features*, Identificar Requisitos Não Funcionais e Escrever Histórias ocorre essa identificação por meio das técnicas de elicitación definidas na seção 6.1, que são: Análise Documental, *Workshop* de Requisitos e Entrevistas.

Na prática específica **Desenvolver Requisitos de Cliente** as necessidades levantadas são convertidas em requisitos documentados, denominados Requisitos de Cliente. Nas atividades Estabelecer Tema de Investimento, Levantar Épicos, Levantar *Features*, Escrever Histórias e Especificar Histórias ocorre a especificação dos requisitos levantados. A atividade de Especificar Histórias atende bem à essa prática, pois é a especificação das histórias levantadas na atividade de Escrever Histórias e também contém os critérios de aceitação que atende a subprática de definir estratégia de verificação e validação.

Na prática específica **Estabelecer uma Definição da Funcionalidade Requerida** o objetivo é analisar as funcionalidades para identificar ações, sequências, entradas e saídas. Na atividade Especificar histórias essa prática é atendida pois há a especificação das histórias e a escrita dos testes de aceitação que proporcionam essa visão das ações, entradas e saídas nas funcionalidades.

Na prática específica **Analisar Requisitos** o objetivo é analisar os requisitos para determinar se estão alinhados com os requisitos de mais alto nível, se estão de acordo com o que o cliente estabeleceu. Nas atividades Fazer reunião de validação dos Épicos, Fazer reunião de validação dos *Features* e Realizar Reunião de Revisão e Retrospectiva da Iteração os requisitos levantados e especificados são validados para confirmar se atendem às necessidades dos clientes.

Na prática específica **Validar Requisitos** o objetivo é garantir que os requisitos identificados e documentados estejam de acordo com o requerido. Nas atividades Fazer reunião de validação dos Épicos, Fazer reunião de validação dos *Features* e Realizar Reunião de Revisão e Retrospectiva da Iteração essa prática é atendida, pois se tratam de reuniões para validações dos requisitos nos diferentes graus de abstração. Todavia, na Reunião de Revisão e Retrospectiva da Iteração a validação é feita na implementação dos requisitos.

6 Elicitação

A elicitação é uma das atividades fundamentais da Engenharia de Requisitos e consiste no processo de identificar itens de informação que determinam as características de um sistema. (JITNAH; HAN; STEELE, 1995)

Para Paetsch, Eberlein e Maurer (2003) a elicitação de requisitos tem o intuito de descobrir os requisitos e identificar as fronteiras do sistema, estas definem o contexto, consultando os *stakeholders* (cliente, usuários e desenvolvedores).

A atividade de elicitação pode ser realizada através da aplicação de várias técnicas, como: Grupos focais, entrevistas, questionários, introspecção, análise de protocolo, prototipagem, animação, análise de cenário, estudo etnográfico, observação, análise de tarefas, *workshops* e *brainstorming*. ((JITNAH; HAN; STEELE, 1995) ; (ZOWGHI; COULIN, 2005))

6.1 Técnicas de Elicitação de Requisitos

As técnicas de elicitação de requisitos foram selecionadas de acordo com a interação entre a equipe e o cliente e de acordo com os níveis de conhecimento e de abstração dos requisitos a serem obtidos. As técnicas escolhidas foram:

- Análise documental;
- *Workshop* de Requisitos;
- Entrevistas.

A técnica de análise documental será empregada inicialmente, pois permite a visualização de formulários, fichas, relatórios e outros documentos da empresa, podendo auxiliar no entendimento do contexto. (FALBO, 2012).

Serão realizados um ou mais *Workshops* para levantamento de requisitos em mais alto nível, de acordo com a necessidade de entendimento da equipe e disponibilidade do cliente. Segundo Nuseibeh e Easterbrook (2000), essa técnica explora a dinâmica em grupo, dessa forma, possibilitando conhecer uma visão geral sobre o problema.

Nos *Workshops* serão atribuídas responsabilidades as pessoas, como a de controlar o tempo da reunião, de fazer anotações e de verificar se a reunião está perdendo o foco,

assim como sugerido por (FALBO, 2012).

Para condução da elicitação nesta técnica, serão considerados os procedimentos relatados por Alexander e Beus-Dukic (2009 apud Falbo (2012)). Dessa forma, as ideias e as necessidades relatadas serão esboçadas em quadro branco para que todos os participantes possam visualizar e opinar sobre as prioridades, os objetivos e diferentes soluções. E por fim um relatório deve ser gerado com todas as informações levantadas no *Workshop*.

Serão realizadas entrevistas para elicitação de requisitos, visto que esse tipo de técnica permite uma compreensão das necessidades e dos requisitos em visões mais específicas, e em um menor nível de abstração que nos *Workshops*. De acordo com Zowghi e Coulin (2005), a entrevista é uma forma rápida de coletar uma grande quantidade de dados. No contexto deste trabalho é uma boa técnica dada a interação que foi estabelecida com o cliente.

As entrevistas podem ser fechadas, abertas ou mistas. As entrevistas fechadas possuem perguntas predefinidas. Já nas entrevistas abertas as perguntas são criadas de acordo com os tópicos explorados. As entrevistas mistas combinam os dois tipos, ou seja, algumas perguntas são definidas anteriormente e de acordo com os assuntos levantados nas reuniões novas perguntas são criadas. (SOMMERVILLE, 2007).

Serão realizadas entrevistas mistas, pois como esta técnica será combinada com outras duas, algumas perguntas podem ser predefinidas baseadas nas elicitações realizadas anteriormente e há a possibilidade de criação de novas perguntas.

As entrevistas serão registradas por meio de anotações, embora seja um processo mais lento, favorece uma análise momentânea dos requisitos levantados e assim auxilia na exploração dos tópicos a serem abordados na entrevista.

7 Tópicos de Gerenciamento de Requisitos

Nesse capítulo estão definidos os atributos que serão acompanhados nos requisitos e a estratégia de rastreabilidade.

7.1 Atributos de Requisitos

De acordo com o [Guia OPEN UP \(2015\)](#), atributos dos Requisitos são as propriedades de um requisito. Assim como uma entidade qualquer no contexto de desenvolvimento de software possui seus atributos, um requisito também possui os seus.

De maneira semelhante a uma entidade do tipo pessoa que possui atributos, tais como idade, cor do cabelo e sexo, cada requisito possui uma origem, uma importância relativa e a data em que foi criado, por exemplo. ([GUIA OPEN UP BASIC, 2009](#)).

De acordo com o [Guia OPEN UP BASIC \(2009\)](#), os atributos são uma fonte muito importante de informações sobre os requisitos e têm a intenção de capturar informações adicionais de cada requisito. Além de apenas definir as propriedades de um requisito, se bem definidos, os atributos podem fornecer significantes informações sobre o estado do desenvolvimento de um sistema. Estabelecendo um paralelo com uma consulta em um banco de dados em que se deseja encontrar todos os homens com cabelo castanho e idade maior que 30 anos, é possível consultar os atributos de um requisito para encontrar os requisitos de alta prioridade para o cliente nos últimos 30 dias. ([GUIA OPEN UP BASIC, 2009](#)).

Com o objetivo de acompanhar e gerir da melhor maneira possível os requisitos, foram estabelecidos os seguintes atributos de requisitos:

- **Data de entrega:** Data em que o requisito deve ser fornecido. Os valores deste atributo serão as próprias datas planejadas de entrega do requisito.

Esse atributo foi definido por fornecer de maneira rápida a previsão de entrega do requisito. Dessa maneira, permite uma visão ampla do planejamento de requisitos, auxiliando nas ações de gerenciamento dos requisitos.

- **Dificuldade:** Uma indicação do nível de esforço necessário ou quão difícil será implementar o requisito. Este atributo pode assumir os valores: alta, média ou baixa.

Esse atributo foi escolhido em razão de informar o quão difícil será implementar um requisito, fornecendo dados que auxiliam no planejamento e organização dos esforços para conclusão das atividades de desenvolvimento de requisitos e para alcançar os resultados esperados.

- Status: Grau de completude, ou seja, o progresso da implementação de um requisito. Este atributo pode adquirir os seguintes valores: completo, parcialmente concluído ou não Iniciado.

Esse atributo foi definido com o intuito de fornecer, a qualquer momento do desenvolvimento dos requisitos, a situação atual do requisito, em relação ao andamento de sua implementação. Ele permite que se tenha uma visão total do andamento do projeto, a partir de cada requisito.

- Prioridade: Declaração da importância relativa do requisito para os *Stakeholders* (envolvidos ou interessados no desenvolvimento do software). Pode receber os seguintes valores: alta, média ou baixa.

Esse atributo foi escolhido em razão de identificar o quão importante um requisito é para os envolvidos no projeto. Ele permite priorizar e planejar a implementação dos requisitos de acordo com suas prioridades, priorizando os requisitos de prioridade mais alta.

7.2 Rastreabilidade de Requisitos

A rastreabilidade pode ser dividida em pré-rastreabilidade e pós-rastreabilidade. A pré-rastreabilidade consiste no rastreamento de informações anteriores a especificação de requisitos. Como por exemplo, a fonte dos requisitos que podem ser: *stakeholders*, documentos e regras de negócio. Já a pós-rastreabilidade está relacionada com os requisitos depois de serem especificados e está mais focada na solução. (DAHLSTEDT; PERSSON, 2005)

A rastreabilidade também pode ser horizontal ou vertical. A horizontal está relacionada com as informações de uma mesma fase do processo de desenvolvimento e a vertical com informações entre várias fases do processo. (DAHLSTEDT; PERSSON, 2005)

De acordo com Paetsch, Eberlein e Maurer (2003), a rastreabilidade auxilia na gerência de mudanças, pois estabelece um relacionamento entre os requisitos, o projeto e a implementação do sistema. Nuseibeh e Easterbrook (2000) afirmam que a rastreabilidade é o coração do gerenciamento de requisitos.

No SAFe os requisitos possuem quatro graus diferentes de abstração dos requisitos: tema de investimento, épicos, *features* e histórias. A partir disso, a rastreabilidade definida para esse trabalho pode ser vista na Figura 6. Será realizada uma rastreabilidade, através da ferramenta de gerenciamento de requisitos, vertical e horizontal. Assim acompanhando a origem das histórias, ou seja, de qual *feature* e épico é derivada e também monitorar as dependências entre as histórias, *features* e épicos.

Também será realizada a pré-rastreabilidade para identificar a fonte dos requisitos. A rastreabilidade definida será feita através da ferramenta de gerenciamento de requisitos.

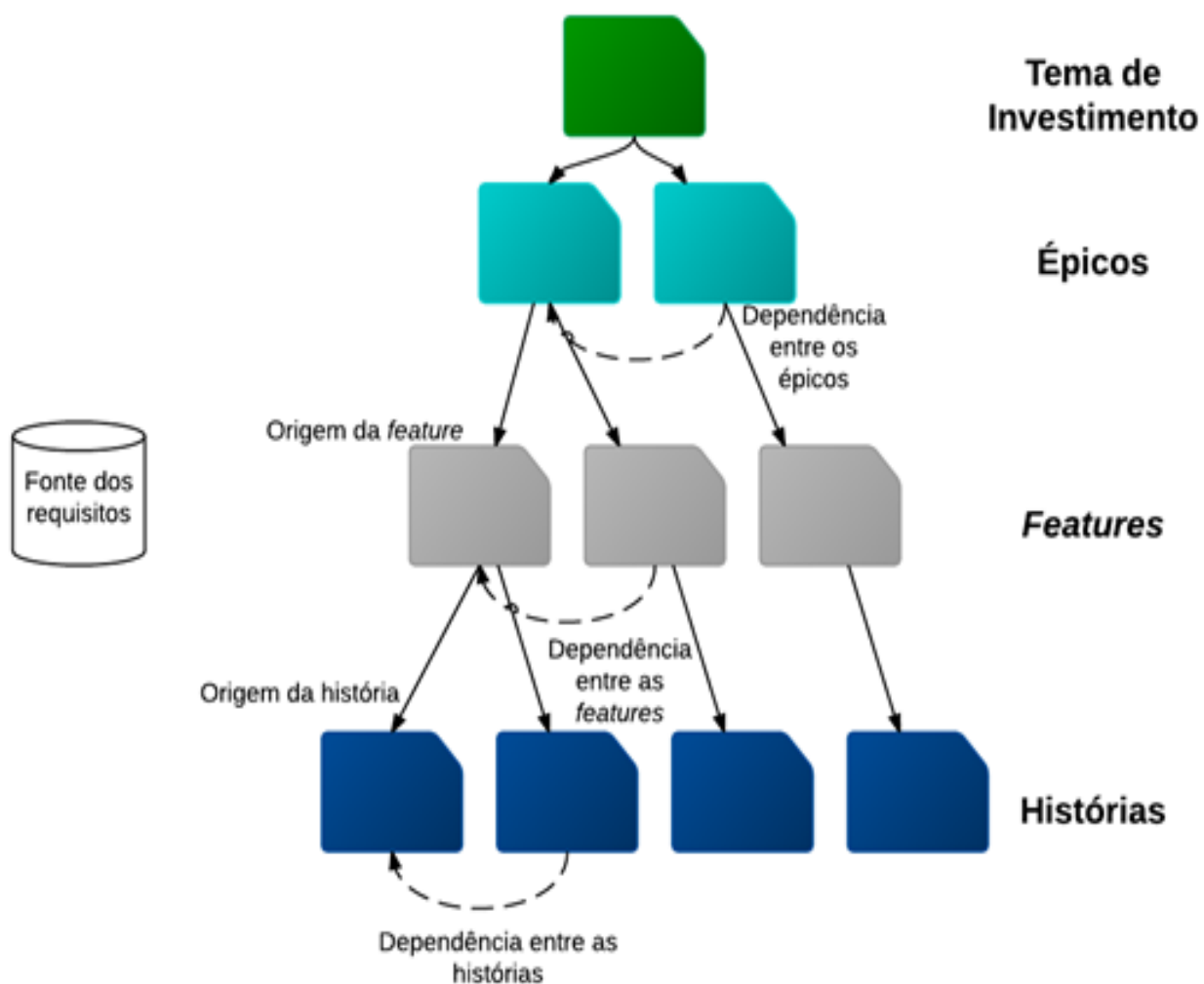


Figura 6 – Estratégia de rastreabilidade

8 Planejamento

O cronograma, é um documento fundamental em qualquer projeto, seja ele pessoal ou um projeto de equipe. Ele consiste na formalização das atividades que se busca realizar desde o início do projeto até seu término. Além disso, o cronograma serve para orientar os envolvidos no projeto no que se refere ao que deverá ser feito no tempo atual.

Atualmente, o cronograma dos alunos da equipe de Requisitos, está integrado com o da equipe de Modelagem de Processos, de modo que o contexto a ser trabalhado se encaixe da melhor forma possível com a metodologia escolhida.

Tendo isso em mente, a ferramenta escolhida para construção do cronograma foi o Gantter que se trata de uma aplicação *web* vinculada ao Google Drive que possibilita além da criação de uma ordem de execução de tarefas, a definição dos responsáveis dessas tarefas e a visualização do grau de conclusão por meio de porcentagem.

Neste primeiro ponto do cronograma, as atividades consistiam em fundamentar conceitos referentes a Engenharia de Requisitos de modo que as melhores escolhas fossem tomadas para realização das tarefas com excelência.









		Name	Duration	Start	Finish	Predecessors	Resources
1		☐ Projeto Matriz Engenharia de Energia	94.38d?	25/03/2015	27/06/2015		Equipe
2		☐ Entrega 1	94.38d?	25/03/2015	27/06/2015		Equipe
3		☐ Engenharia de Requisitos (REQ)	94.38d?	25/03/2015	27/06/2015		Requisitos
4		Reunir equipe de requisitos para apresentação inicial	0.33d?	25/03/2015	25/03/2015		Requisitos
5		Criar cronograma de atividades ER	7d?	06/04/2015	13/04/2015		Hugo Martins
6		Reuniões de alinhamento	32d?	01/04/2015	03/05/2015		Emilie Morais
7		☐ Desenvolver Relatório	70d?	18/04/2015	27/06/2015		Emilie Morais
8		☐ Definir abordagem do processo de engenharia	5d?	19/04/2015	24/04/2015		Emilie Morais
9		Definir conceitos referentes a abordagem de desen	5d?	19/04/2015	24/04/2015		Emilie Morais
10		Justificar a escolha da abordagem	2d?	22/04/2015	24/04/2015		Emilie Morais
11		☐ Definir os processos da engenharia de requisi	58.67d?	30/04/2015	27/06/2015		Filipe Ribeiro
12		Definir atividades do processo de engenharia de	1d?	26/06/2015	27/06/2015		Filipe Ribeiro
13		Definir papéis	1.67d?	01/05/2015	02/05/2015		Emilie Morais
14		Definir artefatos	1.67d?	30/04/2015	01/05/2015		Emilie Morais
15		Modelar processo definido no Bizagi	0.33d?	02/05/2015	02/05/2015		Emilie Morais
16		Validar o processo da engenharia de requisitos	7d?	01/05/2015	08/05/2015		Emilie Morais
17		☐ Definir as práticas que serão utilizadas dos m	5d?	26/04/2015	01/05/2015		Edson Gomes
18		Mapear as práticas com os processos da engen	5d?	26/04/2015	01/05/2015		Edson Gomes

Figura 7 – Primeira Parte do Cronograma

Neste segundo ponto do cronograma, basicamente busca-se um complemento do que já foi especificado no primeiro ponto, porém, agora as atividades a serem desenvolvidas

necessitam que as outras estejam concluídas.

		Name	Duration	Start	Finish	Predecessors	Resources
19		<input type="checkbox"/> Definir técnicas de elicitação	3.33d?	28/04/2015	01/05/2015		Emilie Morais
20		Justificar técnicas de elicitação que foram definidas	3.33d?	28/04/2015	01/05/2015		Emilie Morais
21		Definir estratégia de rastreabilidade dos requisitos	3.33d?	28/04/2015	01/05/2015		Emilie Morais
22		Especificar atributos dos requisitos	3.33d?	28/04/2015	01/05/2015		Edson Gomes
23		<input type="checkbox"/> Definir ferramenta para gestão de requisitos	6d?	18/04/2015	24/04/2015		Hugo Martins
24		Selecionar três ferramentas para gestão de requisitos	6d?	18/04/2015	24/04/2015		Hugo Martins
25		Avaliar três ferramentas para gestão de requisitos	6d?	18/04/2015	24/04/2015		Edson Gomes
26		Comparar as ferramentas para gestão de requisitos	6d?	18/04/2015	24/04/2015		Filipe Ribeiro
27		Escolha da ferramenta para gestão de requisitos	6d?	18/04/2015	24/04/2015		Filipe Ribeiro
28		<input type="checkbox"/> Modelagem de Processos (MPR)	31.38d?	02/04/2015	03/05/2015		Modelagem
29		<input type="checkbox"/> Planejar Projeto	18d?	02/04/2015	20/04/2015		
30		Planejar atividades	1.38d?	02/04/2015	03/04/2015		Augusto Modesto
31		Reunir equipes para apresentação inicial	0.38d?	03/04/2015	03/04/2015		Augusto Modesto
32		Criar cronograma de atividade de MPR	5.38d	05/04/2015	10/04/2015	30	Fábio Barbosa
33		<input type="checkbox"/> Selecionar ferramenta de BPMS	8.01d?	12/04/2015	20/04/2015		
34		Avaliar Bizagi	6.38d?	12/04/2015	18/04/2015		Fábio Barbosa
35		Avaliar Bonita	6.38d?	12/04/2015	18/04/2015		Pedro Felipe
36		Avaliar Intália	6.38d?	12/04/2015	18/04/2015		Augusto Modesto

Figura 8 – Segunda Parte do Cronograma

		Name	Duration	Start	Finish	Predecessors	Resources
37		Gerar relatório de avaliação	1d?	19/04/2015	20/04/2015	34,35,36	Augusto Modesto
38		<input type="checkbox"/> Mapear Processo Atual	9d?	16/04/2015	25/04/2015		
39		<input type="checkbox"/> Entender o negócio	3.38d?	16/04/2015	19/04/2015		
44		Avaliar entendimento de negócio (reunião)	0.38d?	21/04/2015	21/04/2015	39	Fábio Barbosa
45		<input type="checkbox"/> Gerar 1ª versão do modelo AS-IS	1.38d?	22/04/2015	23/04/2015		
46		Modelar 1ª versão AS-IS	0.38d?	22/04/2015	22/04/2015	44	Pedro Felipe
47		Validar modelagem	0.38d?	23/04/2015	23/04/2015	46	Augusto Modesto
48		<input type="checkbox"/> Gerar 2ª Versão do modelo AS-IS	0.76d?	24/04/2015	25/04/2015		
49		Modelar 2ª versão AS-IS	0.38d?	24/04/2015	24/04/2015	47	Fábio Barbosa
50		Validar modelagem	0.38d?	24/04/2015	25/04/2015	49	Pedro Felipe
51		<input type="checkbox"/> Analisar Processo Atual	8.39d?	25/04/2015	03/05/2015		
52		Priorizar macro processos	0.38d?	26/04/2015	26/04/2015	41	Augusto Modesto
53		Identificar regras de negócio	0.38d?	27/04/2015	27/04/2015	52	Fábio Barbosa
54		Simular última versão do AS-IS	2.38d?	25/04/2015	27/04/2015	38	Pedro Felipe
55		Identificar os pontos críticos	1d?	27/04/2015	28/04/2015	54	Augusto Modesto
56		Levantar melhorias para os pontos críticos identificados	1d?	28/04/2015	29/04/2015	55	Fábio Barbosa
57		Documentar atividades do processo	3.38d?	30/04/2015	03/05/2015	54	Pedro Felipe
58		<input type="checkbox"/> Atividades Conjuntas (REQ e MPR)	7.38d?	26/04/2015	03/05/2015		Equipe

Figura 9 – Terceira Parte do Cronograma

Com as atividades definidas e documentadas, neste ponto do projeto busca-se o

preparo de uma primeira apresentação como forma de visualização prévia do que de fato será realizado e como, além do começo das atividades referentes a segunda entrega.



















		Name	Duration	Start	Finish	Predecessors	Resources
59		<input type="checkbox"/> Preparar a apresentação do trabalho	0.38d?	03/05/2015	03/05/2015		Edson Gomes
60		Criar slide de apresentação	0.38d?	03/05/2015	03/05/2015		Edson Gomes, Au
61		Organizar a apresentação	0.38d?	03/05/2015	03/05/2015		Emilie Moraes, Pe
62		Integrar cronogramas (grupo 4)	0.33d	26/04/2015	26/04/2015		Augusto Modesto
63		Identificar desenvolvimento de solução (ferramental)	0.38d?	26/04/2015	26/04/2015		Augusto Modesto
64		<input type="checkbox"/> Entrega 2	55.84d?	02/05/2015	27/06/2015		Requisitos
65		<input type="checkbox"/> Engenharia de Requisitos (REQ)	42.63d?	13/05/2015	25/06/2015		Requisitos
66		<input type="checkbox"/> Desenvolver relatório	42.63d?	13/05/2015	25/06/2015		Emilie Moraes
67		Apresentar o contexto de negócio	2d?	13/05/2015	15/05/2015		Emilie Moraes
68		<input type="checkbox"/> Levantar Épicos	6.33d?	16/05/2015	22/05/2015		Filipe Ribeiro
69		Especificar Épicos	5d?	16/05/2015	21/05/2015		Filipe Ribeiro
70		Fazer reunião de validação dos épicos	0.33d?	22/05/2015	22/05/2015		Filipe Ribeiro
71		<input type="checkbox"/> Levantar features	5.33d?	23/05/2015	28/05/2015		Edson Gomes
72		Especificar features	5d?	23/05/2015	28/05/2015		Edson Gomes
73		Fazer reunião de validação das features	0.33d?	28/05/2015	28/05/2015		Edson Gomes
74		Construir documento de visão	3d?	29/05/2015	01/06/2015		Hugo Martins
75		Construir roadmap	2d?	01/06/2015	03/06/2015		Emilie Moraes
76		Registrar a experiência de execução das técnicas d	2d?	03/06/2015	05/06/2015		Hugo Martins

Figura 10 – Quarta Parte do Cronograma

Dessa forma, as atividades definidas antes da entrega da primeira parte do projeto, mais precisamente na definição do projeto, serão de fato implementadas de forma prática. Haverá reuniões com a equipe de modelagem para alinhamento de trabalhos, conversas a fim de negociar novas funcionalidades.

		Name	Duration	Start	Finish	Predecessors	Resources
77		<input type="checkbox"/> Planejar Release	3d?	22/06/2015	25/06/2015		Emilie Moraes
78		Priorizar features	1d?	22/06/2015	23/06/2015		Edson Gomes
79		Escrever histórias	1d?	23/06/2015	24/06/2015		Hugo Martins
80		Especificar histórias da primeira iteração	1d?	24/06/2015	25/06/2015		Filipe Ribeiro
81		<input type="checkbox"/> Executar iteração	23d?	26/05/2015	18/06/2015		Filipe Ribeiro
82		Alocar histórias	23d?	26/05/2015	18/06/2015		Emilie Moraes
83		Executar iteração	22d?	27/05/2015	18/06/2015		Hugo Martins
84		Fazer reunião de retrospectiva	17d?	01/06/2015	18/06/2015		Edson Gomes
85		Especificar histórias da próxima iteração	23d?	26/05/2015	18/06/2015		Emilie Moraes
86		Registrar os requisitos e a rastreabilidade na ferram	3.67d?	26/05/2015	29/05/2015		Emilie Moraes
87		Relatar experiência da execução do trabalho	0.33d?	19/06/2015	19/06/2015		Filipe Ribeiro
88		Relatar experiência da disciplina de Engenharia de	0.33d?	19/06/2015	19/06/2015		Filipe Ribeiro
89		<input type="checkbox"/> Desenvolver solução de software	16d?	03/06/2015	19/06/2015		Emilie Moraes
90		<input type="checkbox"/> Definir a linguagem de programação do softw	2d?	03/06/2015	05/06/2015		Emilie Moraes
91		Estudar a linguagem de programação definida	2d?	03/06/2015	05/06/2015		Emilie Moraes
92		<input type="checkbox"/> Definir ambiente de desenvolvimento	2d?	03/06/2015	05/06/2015		Filipe Ribeiro
93		Estudar o ambiente de desenvolvimento definido	2d?	03/06/2015	05/06/2015		Filipe Ribeiro
94		<input type="checkbox"/> Definir controle de versão	2d?	03/06/2015	05/06/2015		Hugo Martins

Figura 11 – Quinta Parte do Cronograma

Por fim, a solução técnica será apresentada, e para isso é necessário primeiro de uma detalhação de como será alcançada essa solução, codificação ou via ferramentas BPM. Feito isso o grupo estará pronto para a apresentação final.
















		Name	Duration	Start	Finish	Predecessors	Resources
94		 Definir controle de versão	2d?	03/06/2015	05/06/2015		Hugo Martins
95		Criar repositório	2d?	03/06/2015	05/06/2015		Hugo Martins
96		Desenvolver solução que atenda os requisitos	5.33d?	14/06/2015	19/06/2015		Emilie Morais
97		 Preparar a apresentação do trabalho	2d?	17/06/2015	19/06/2015		Filipe Ribeiro
98		Organizar a apresentação	2d?	17/06/2015	19/06/2015		Filipe Ribeiro
99		Criar slide de apresentação	2d?	17/06/2015	19/06/2015		Emilie Morais
100		Apresentar a solução de software referente aos requisitos	2d?	17/06/2015	19/06/2015		Emilie Morais
101		 Modelagem de Processos (MPR)	1d?	26/06/2015	27/06/2015		
120		 Atividades Conjuntas (REQ e MPR)	55.84d?	02/05/2015	27/06/2015		
121		 Preparar a apresentação do trabalho	55.84d?	02/05/2015	27/06/2015		Edson Gomes
122		Criar slide de apresentação	0.84d?	26/06/2015	27/06/2015		Edson Gomes, Au
123		Organizar a apresentação	0.84d?	02/05/2015	03/05/2015		Emilie Morais, Pe

Figura 12 – Sexta Parte do Cronograma

9 Ferramentas de Gerenciamento de Requisitos

Neste capítulo serão discutidas e analisadas três ferramentas para gerenciamento de requisitos para que seja escolhida uma, entre elas, que se adeque melhor ao processo de Engenharia de Requisitos proposto neste trabalho.

9.1 Ferramentas encontradas

Ao longo dos últimos anos, tem havido uma mudança dramática no gerenciamento de requisitos, nas ferramentas disponíveis no mercado, e seus recursos disponíveis. (BE-ATTY, 2013)

Foram encontradas diversas ferramentas tais como: Jeremia, Open Source Requirements Management Tool (OSRMT), Xuse, Tiger Pro, TraceCloud e Caliber. Considerando o critério de melhor atender as necessidades do processo de engenharia de requisitos atual, foram selecionadas, entre essas, apenas três para a avaliação. Essas ferramentas estão detalhadas a seguir.

9.1.1 Tiger Pro

O Tiger Pro (Tool to InGest and Elucidate Requirements PROfessional) é uma ferramenta de gerenciamento de requisitos licenciada para o uso educacional que permite que os requisitos, além de também poderem ser adicionados diretamente no programa, possam ser adicionados através de documentos.

“Produz um sumário de informações na elucidação de requisitos, alocação de prioridades, definição de riscos e estimativa de custo em formato de texto ou gráfico.” (TEIXEIRA, 2009)

Apesar do software ser licenciado para o uso educacional, é possível utilizá-lo para fins comerciais, desde que haja uma autorização do responsável pelo software.

9.1.2 TraceCloud

De acordo com informações obtidas no site da ferramenta, o TraceCloud é um software para gerência de requisitos independente se o projeto possui abordagem adaptativa ou se possui abordagem tradicional. Esse software gerencia as mudanças dos requisitos, desde mudanças de baixo nível até as de mais alto nível. Além disso, o TraceCloud dá um suporte completo de rastreabilidade dos requisitos.

O TraceCloud é bastante flexível ou seja, ele se adapta a qualquer processo de negócio. Vale ressaltar que dentro do contexto de nosso projeto, onde os atributos de requisitos foram outrora definidos, o TraceCloud fornece suporte a cada atributo.

O TraceCloud não é gratuito, mas possui versão gratuita para testes.

9.1.3 Caliber

Segundo informações obtidas pelo site da ferramenta, o Caliber possui uma solução completa para gerenciamento de requisitos que garante a conformidade e alinhamento de desenvolvimento para atender as necessidades de negócios. Ele facilita a colaboração dos *stakeholders*, possui uma visualização rica, gerenciamento robusto e uma rastreabilidade dos requisitos bem definida e completa.

O Caliber garante a entrega de um software com maior precisão e que atenda as necessidades dos clientes.

O software permite a participação do usuário de forma direta com o processo aumentando a clareza e precisão. Com isso, o usuário dá feedbacks para garantir que os requisitos não sejam definidos de forma inequívoca.

O Caliber fornece uma rastreabilidade completa dos requisitos para modelar as relações internas e externas dos mesmos. Essa rastreabilidade suporta as mudanças dos requisitos e responde como essas mudanças causam impacto no projeto.

O Caliber não é gratuito, mas possui versão de 30 dias para testes.

9.2 Análise das Ferramentas

Foram instaladas e testadas cada uma das ferramentas pesquisadas para que fosse decidido qual das três se adequasse melhor ao processo.

9.2.1 Tiger Pro

A instalação da ferramenta pode ser feita de maneira bem trivial, no entanto é necessária uma base de dados para que a ferramenta funcione. Essa necessidade pode ser suprida por meio do uso de um arquivo do tipo CSV ou utilizando uma base de dados MySQL ou Oracle, por exemplo.

A ferramenta tem quase 10 anos de lançamento, sem novas atualizações desde 2006. Possui interface um pouco confusa, mas aparentemente funcional. Dentre aos dados relacionadas com os requisitos, os mais importantes que a ferramenta permite o gerenciamento são:

- Critérios de Aceitação;
- Análise Relacional ou Base Lógica;
- Rastreabilidade;
- Prioridade;
- Risco;
- Estimativa de Custos.

Dentre os atributos de requisitos que foram definidos para controlar e informar sobre os requisitos no projeto, apenas Prioridade foi encontrado na ferramenta. A ferramenta permite o gerenciamento de outros atributos, porém estes outros não foram definidos previamente e nem observamos vantagens no seu uso para o projeto atual. Por causa da limitação da ferramenta em relação aos atributos que ela controla e os atributos que escolhemos, e por ser uma ferramenta já descontinuada, a ferramenta não se adequa ao processo definido.

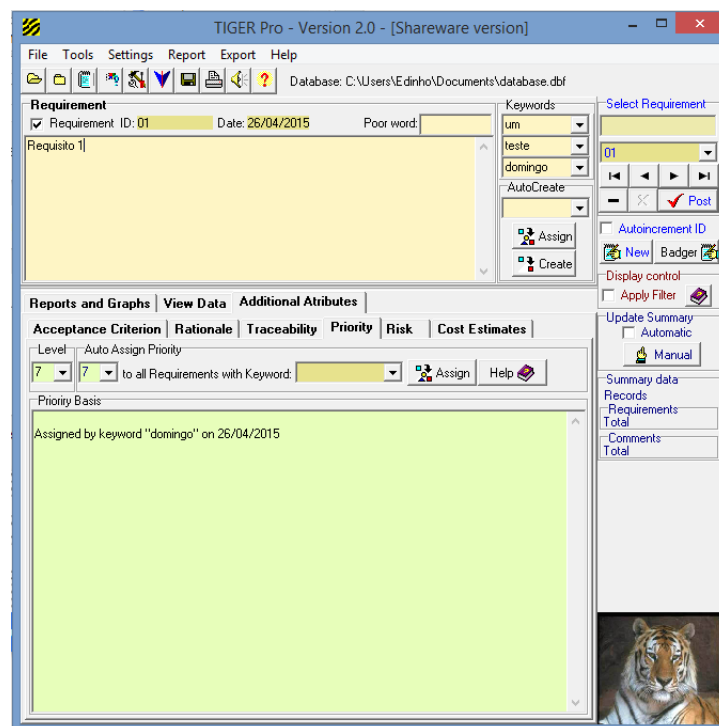


Figura 13 – Tela inicial da ferramenta Tiger-Pro

9.2.2 TraceCloud

Não há grandes problemas no que se refere a instalação da ferramenta, pois ela é *online*. A ferramenta pode ser usada por um grupo de usuários e os dados da mesma

podem ser alterados por todos os envolvidos no projeto. Uma das grandes desvantagens da ferramenta, se refere ao fato da mesma ser via *Web*, ou seja, em locais em que não há acesso a internet, é impossível de se trabalhar.

A ferramenta vai de encontro com o que o grupo definiu como atributos de estudo, pois ela registra perfeitamente e de forma intuitiva as características dos requisitos que são gerenciadas pela mesma, algumas delas são:

- Gerência de mudanças;
- Status de aprovação dos requisitos;
- Status de andamento das atividades envolvendo os requisitos;
- Prioridade no desenvolvimento dos requisitos;
- Requisitos relacionados.

O TraceCloud, fornece aos usuários uma divisão muito bem definida dos requisitos funcionais e dos de negócio por meio de pastas no canto esquerdo. Essas mesmas pastas possuem também uma área direcionada aos casos de testes que podem ser devidamente documentadas em um devido espaço. Além disso, há a possibilidade de dividir algumas funções para membros do projeto e definir a data de entrega da mesma, o que é de grande valor para qualquer projeto, criando uma espécie de agenda da equipe para as mais diversas iterações. Há a possibilidade do rastreamento do requisito desde sua criação, até o mesmo ser desenvolvido por completo, além da possibilidade de vinculá-lo a um outro requisito.

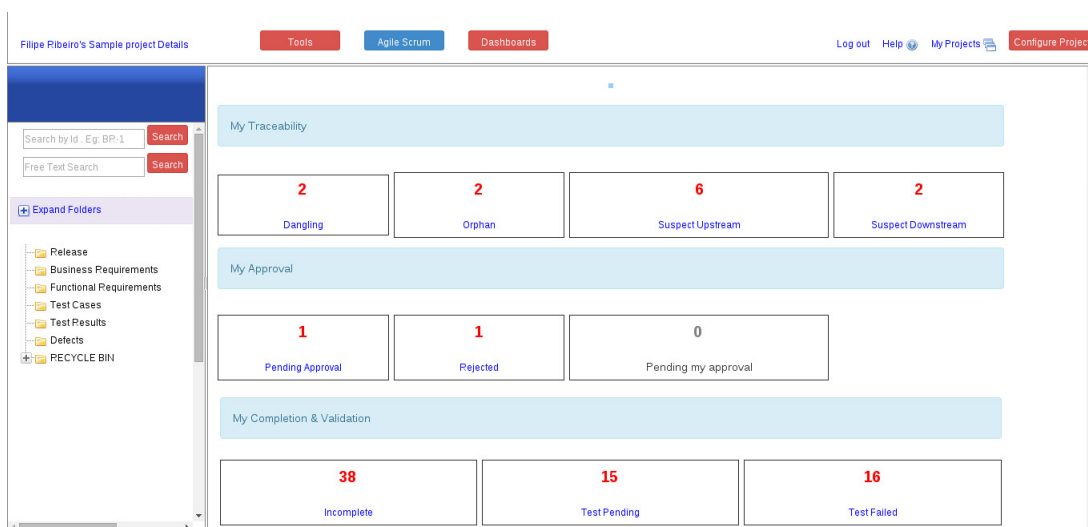


Figura 14 – Tela inicial da ferramenta TraceCloud

9.2.3 Caliber

A instalação da ferramenta é fácil, sendo necessário fazer um cadastro. A ferramenta é paga, mas possui a versão para testes por 30 dias. Foi escolhida essa versão para avaliar a ferramenta.

A ferramenta gerencia muito bem os requisitos nos seguintes critérios:

- Mudança dos requisitos;
- Prioridade dos requisitos;
- Rastreabilidade;
- Relacionamento dos requisitos;
- Análise dos requisitos.

Qualquer alteração que os requisitos sofrem é registrada pela ferramenta no histórico de mudanças a data e hora da mudança, o responsável e há também a possibilidade de se fazer um comentário referente a tal mudança.

A organização dos itens da ferramenta usa um padrão já conhecido pelos usuários de outras ferramentas similares, barra a esquerda com os dados do projeto, menu na barra superior. Isso torna a ferramenta fácil de usar, evitando que o usuário fique confuso.

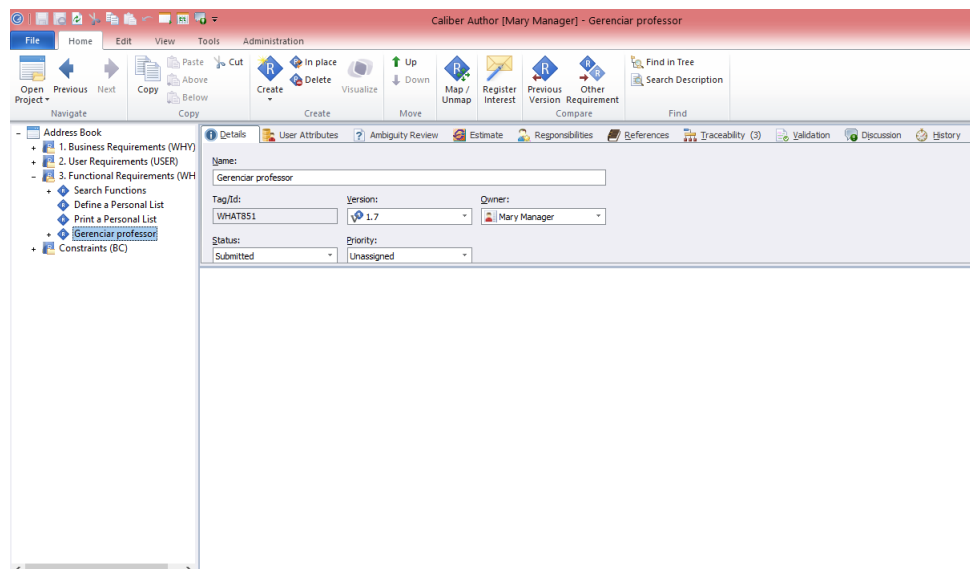


Figura 15 – Tela inicial da ferramenta Caliber

9.3 Critérios de Avaliação das Ferramentas

De acordo com Beatty (2013) os critérios para avaliar uma ferramenta são: Armazenamento dos requisitos e de seus atributos

- Artefatos que estão relacionados
- Gestão de mudanças
- Rastreabilidade dos requisitos
- Flexibilidade
- Usabilidade
- Adaptação ao contexto

a partir disso, foram definidos os critérios para avaliar as ferramentas nesse trabalho.

- Armazenamento dos requisitos e de seus atributos: nesses pontos, o foco da avaliação era descobrir a capacidade da ferramenta em guardar informações sobre os requisitos. Visto que há a possibilidade da mesma alterar ou simplesmente apagar dados inseridos, a partir de testes e do manual da ferramenta foi possível descobrir se com o uso da mesma, estava-se livre de tal risco.
- Artefatos que estão relacionados: nessa avaliação, buscou-se compreender e definir quais as relações presentes em dado requisito, ou seja, se o mesmo se relaciona com outros requisitos e se ao fim de dada iteração, algum documento é gerado.
- Gestão de mudanças: buscou-se avaliar o comportamento da ferramenta no que diz respeito as mudanças que o requisito pode sofrer ao longo do projeto. A importância de se avaliar isso tange a questão da qualidade do que foi proposto, se de fato é pouco mutável e menos suscetível a falhas.
- Rastreabilidade dos requisitos: buscou-se a avaliação para saber se a rastreabilidade dos requisitos é realizada de maneira satisfatória, fornecendo características essenciais como origem dos requisitos em seus diferentes níveis de abstração e dependências entre os requisitos.
- Flexibilidade: neste critério, buscou-se avaliar a capacidade que a ferramenta tem de se adequar aos projetos. Se a ferramenta suporta desde um projeto menor até um projeto mais complexo.
- Usabilidade: com essa questão, buscou-se avaliar as facilidades encontradas no manuseio da ferramenta, se ela possuía nomes significativos, se a mesma é executada de maneira fluída, sem muitos travamentos, se a interface era fácil de usar e se era fácil de se lembrar como se usa.
- Adaptação ao contexto: nesse ponto, fez-se uma verificação e análise a fim de descobrir se a ferramenta oferecia um suporte considerável e único para a abordagem definida pelo time e para o processo definido.

Após a pesquisa das ferramentas para gestão de requisitos e a análise de cada uma das encontradas, foi feito um quadro comparativo com critérios definidos acima, que pode ser visto na Figura 16 que avalia cada uma das ferramentas. Cada um desses critérios foram avaliados de 0 a 5, sendo:

- 0 - Muito ruim - Não atende de forma alguma ao critério.
- 1 - Ruim - Não atende ao critério da forma esperada.
- 2 - Razoável - Atende ao critério, mas seu retorno não é de forma significativa.
- 3 - Bom - Atende ao critério da forma esperada.
- 4 - Muito bom - Atende ao critério além da forma esperada.
- 5 - Excelente - Atende ao critério além da forma esperada, com um retorno bastante significativo e de qualidade.

Avaliação das ferramentas			
Item/Ferramenta	Tiger Pro	TraceCloud	Caliber
Armazenamento dos requisitos	3	4	4
Armazenamento de atributos dos requisitos	2	4	4
Armazenamento de artefatos relacionados aos requisitos	1	3	4
Relacionamento entre os requisitos	2	5	4
Gestão de mudanças	3	3	3
Rastreabilidade	3	5	4
Flexibilidade	2	3	3
Usabilidade	3	5	3
Para o processo de Engenharia de Requisitos da Empresa Junior de Engenharia de Energia	2	5	5
Tipo	Gratuito	Pago	Pago

Figura 16 – Quadro comparativo entre ferramentas.

9.4 Escolha da Ferramenta

A ferramenta escolhida foi o TraceCloud. Como já dito, o TraceCloud é uma ferramenta *web*, portanto o grupo se compromete e assume os riscos de conseguir apresentar as documentações geradas por ela independente da qualidade da internet em dias de apresentação. A vantagem é que todos componentes do grupo ganham uma maior autonomia com todos podendo alterar os documentos sempre que necessário, pois em casos de ferramentas offline, certamente seria necessário a delegação da função para um único aluno

ficar responsável por gerenciar tais requisitos, o que é um grande risco, visto que o arquivo do computador pode sofrer um dano assim como o próprio sistema local.

Como explicado e avaliado na Figura 16, as ferramentas avaliadas pelo grupo receberam notas de 0 a 5 em algumas características previamente estudadas, e comparando com as outras ferramentas avaliadas, em vários quesitos ela obteve a melhor pontuação. Em especial, no tópico que diz respeito ao processo de Engenharia de Requisitos da Empresa Júnior de Engenharia de Energia, a ferramenta possui uma opção de escolha entre metodologias ágeis e tradicionais, sendo que cada uma recebe um melhor tratamento e as instruções que melhor se adequam ao trabalho. Além de fornecer um suporte ao Scrum, tal ferramenta pode documentar as datas das modificações que os requisitos recebem ao longo do projeto garantindo total rastreabilidade dos mesmos e auxiliando na gerência de mudanças.

Por fim, outro ponto que se deseja utilizar bastante é o delegamento de funções por iteração. Tendo em vista que a cada período de tempo algumas tarefas deverão ser feitas, hora individuais e hora em grupo, para obtenção de um maior controle, organização e para reforço do cronograma (usado com outra ferramenta).

10 Considerações Finais

Perpassadas algumas etapas de busca e construção de conhecimentos sobre os Requisitos, além de estudos sobre a definição de um processo de Engenharia de Requisitos e temas correlacionados ao desenvolvimento de *software*, pôde-se observar o quão ampla é a área de Requisitos, e como vem se aperfeiçoando a cada dia.

Com o objetivo de definir um processo de Engenharia de Requisitos para ser aplicado ao contexto da Empresa Júnior Matriz, foi possível aplicar conhecimentos, metodologias, técnicas e análises na prática, durante o decurso de definição do processo Engenharia de Requisitos. O que possibilitou aos membros das equipes, tanto de Requisitos quanto de Modelagem, uma ganho valioso de conhecimento. Destaca-se aqui, a equipe de Requisitos que trabalhou de forma integrada e esforçada para executar as tarefas e atender aos prazos definidos.

Apesar do ganho de conhecimento, também existiram dificuldades na compreensão de alguns conceitos e técnicas na área de Requisitos, além de contratempos na organização do grupo, definição de reuniões e outros inconvenientes, inerentes ao trabalho em equipe. No entanto, por meio do empenho e dedicação dos integrantes da equipe foi possível vencer esses contratempos e inconvenientes.

Espera-se que o processo de Engenharia de Requisitos definido possa ser aplicado de forma a alcançar os objetivos da Empresa Matriz, agregando valor e resolvendo uma questão problemática da empresa. Além de permitir aos membros da equipe, a vivência na prática do conteúdo visto em sala de aula.

Referências

- ALVES, S. de R.; ALVES, A. L. Engenharia de requisitos em metodologias Ágeis. 2008. Citado na página 9.
- BEATTY, J. *Winning the Hidden Battle: Requirements Tool Selection and Adoption*. [S.l.: s.n.], 2013. Citado 2 vezes nas páginas 39 e 43.
- CALIBER. *Features of Caliber*. 2015. Disponível em: <<http://www.borland.com/Products/Requirements-Management/Caliber>>. Nenhuma citação no texto.
- CARNEGIE MELLON UNIVERSITY. *CMMI for Development, Version 1.3*. 2010. Citado 2 vezes nas páginas 25 e 27.
- DAHLSTEDT Åsa G.; PERSSON, A. Requirements interdependencies: State of the art and future challenges. 2005. Citado na página 32.
- FALBO, R. de A. *Engenharia de Requisitos: Notas de Aula*. 2012. Citado 2 vezes nas páginas 29 e 30.
- GUIA OPEN UP. *Conceito: Atributos de Requisitos*. 2015. Disponível em: <http://www.wthreex.com/rup/openup_15_ptbr/index.html>. Citado na página 31.
- GUIA OPEN UP BASIC. *Conceito: Atributos de Requisitos*. 2009. Disponível em: <http://epf.eclipse.org/wikis/openuppt/openup_basic/guidances/concepts/requirement_attributes,_VQ268O0KEdqHTdbLTmC5IQ.html>. Citado na página 31.
- JITNAH, D.; HAN, J.; STEELE, P. Software requirements engineering: An overview. 1995. Citado 2 vezes nas páginas 5 e 29.
- LEFFINGWELL, D. *Agile Software Development with Verification and Validation in High Assurance and Regulated Environments*. 2011. Citado 3 vezes nas páginas 9, 10 e 11.
- NUSEIBEH, B.; EASTERBROOK, S. *Requiments Engineering: A Roadmap*. 2000. Citado 2 vezes nas páginas 29 e 32.
- PAETSCH, F.; EBERLEIN, D. A.; MAURER, D. F. Requirements engineering and agile software development. 2003. Citado 2 vezes nas páginas 29 e 32.
- PRESSMAN, R. S. *Engenharia de Software: uma abordagem profissional*. [S.l.: s.n.], 2011. Citado 3 vezes nas páginas 5, 9 e 25.
- SHWABER; SUTHERLAND. Guia do scrum. 2013. Citado 2 vezes nas páginas 10 e 11.
- SOFTEX, S. para Promoção da Excelência do S. B. M. P. S. *Guia Geral, “MPS. BR-Melhoria de Processo do Software Brasileiro,”*. 2012. Citado na página 25.
- SOMMERVILLE, I. *Engenharia de Software*. 8. ed. [S.l.: s.n.], 2007. Citado na página 30.

TEIXEIRA, R. A. R. *Estudo comparativo entre ferramentas de Gerência de Requisitos*. 2009. Citado na página 39.

TRACECLOUD. *Profuct features*. 2015. Disponível em: <<https://www.tracecloud.com/GloreeJava2/jsp/WebSite/TCFeatures.jsp>>. Nenhuma citação no texto.

TRACECLOUD. *Requirements Management*. 2015. Disponível em: <<https://www.tracecloud.com/GloreeJava2/jsp/WebSite/TCHome.jsp>>. Nenhuma citação no texto.

VASCO, C. G.; VITHOFT, M. H.; ESTANTE, P. R. C. Comparação entre metodologias rup e xp. 2006. Citado na página 10.

ZOWGHI, D.; COULIN, C. Requirements elicitation: A survey of techniques, approaches, and tools. 2005. Citado 2 vezes nas páginas 29 e 30.