



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

## Relatório do Projeto Fábrica de Massas

Autor: Daniel Moura  
Eduardo Quintino Gomes  
Miguel Nery  
Miguel Pimentel  
Orientador: Elaine Venson

Brasília, DF  
2016





Daniel Moura  
Eduardo Quintino Gomes  
Miguel Nery  
Miguel Pimentel

## **Relatório do Projeto Fábrica de Massas**

Trabalho submetido ao curso de Requisitos  
de Software - 201308da Universidade de Bra-  
sília.

Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA

Orientador: Elaine Venson

Brasília, DF  
2016

---

Daniel Moura  
Eduardo Quintino Gomes  
Miguel Nery  
Miguel Pimentel  
Relatório do Projeto Fábrica de Massas/ Daniel Moura  
Eduardo Quintino Gomes  
Miguel Nery  
Miguel Pimentel. – Brasília, DF, 2016-  
51 p. : il. (algumas color.) ; 30 cm.

Orientador: Elaine Venson

Trabalho de Requisitos de Software – Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA , 2016.

1. Palavra-chave01. 2. Palavra-chave02. I. Elaine Venson. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Relatório do Projeto Fábrica de Massas

CDU 02:141:005.6

---

Daniel Moura  
Eduardo Quintino Gomes  
Miguel Nery  
Miguel Pimentel

## **Relatório do Projeto Fábrica de Massas**

Trabalho submetido ao curso de Requisitos de Software - 201308da Universidade de Brasília.

Trabalho aprovado. Brasília, DF, 01 de junho de 2013:

---

**Elaine Venson**  
Orientador

---

**Titulação e Nome do Professor**  
**Convidado 01**  
Convidado 1

---

**Titulação e Nome do Professor**  
**Convidado 02**  
Convidado 2

Brasília, DF  
2016



## Lista de ilustrações





## Lista de tabelas



# Lista de abreviaturas e siglas

IEEE	Institute of Electrical and Electronics Engineers, Instituto de Engenheiros Eletricistas e Eletrônicos.
SAFe	Scaled Agile Framework, Framework Ágil Escalável
MPsBR	Melhoria de Processo do Software Brasileiro



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>I</b>	<b>JUSTIFICATIVA DA ABORDAGEM</b>	<b>15</b>
<b>2</b>	<b>JUSTIFICATIVA DA ABORDAGEM</b>	<b>17</b>
<b>II</b>	<b>PROCESSO DE ENGENHARIA DE REQUISITOS</b>	<b>23</b>
<b>3</b>	<b>PROCESSO DE ENGENHARIA DE REQUISITOS</b>	<b>25</b>
<b>4</b>	<b>MODELO DE QUALIDADE DE PROCESSO</b>	<b>33</b>
4.0.1	Modelo de Qualidade de processo	33
<b>5</b>	<b>ELICITAÇÃO DE REQUISITOS</b>	<b>39</b>
<b>5.1</b>	<b>Técnicas de Elicitação de Requisitos</b>	<b>39</b>
5.1.1	Observação	40
5.1.2	Entrevista	40
5.1.3	Protótipo	40
5.1.4	Workshop	40
<b>6</b>	<b>GERENCIAMENTO DE REQUISITOS</b>	<b>41</b>
<b>6.1</b>	<b>Rastreabilidade de Requisitos</b>	<b>41</b>
<b>6.2</b>	<b>Atributos dos Requisitos</b>	<b>42</b>
<b>7</b>	<b>FERRAMENTAS DE GESTÃO DE REQUISITOS</b>	<b>45</b>
<b>8</b>	<b>REFERÊNCIAS</b>	<b>49</b>
	<b>REFERÊNCIAS</b>	<b>51</b>



# 1 Introdução

O presente trabalho tem como finalidade apresentar o resultado da Processo de Engenharia de Requisitos de Software para a Fábrica de Massas do Chef Nery. O empreendimento trata-se de uma micro empresa que fabrica diferentes tipos de massas.

O contexto relacionado a necessidade do projeto é principalmente relacionado a necessidade de um melhor gerenciamento de clientes visando então buscar uma melhor forma de gerenciar seus pedidos e divulgar seus produtos.

Para isso, é de fundamental importancia utilizar uma metodologia para o levantamento de requisitos e gerenciamento do processo em questão.

Com uma análise acerca do contexto foi utilizado uma abordagem ágil composto de atividades do Scaled Agile Framework (SAFe). Com isso, foi possível desenhar o Processo de Engenharia de Requisitos contendo um também um modelo do Processo que será implementado no Trabalho 2.

## 1.1 Visão Geral do Relatório

- Introdução: tem como finalidade apresentar um resumo deste documento;
- Contexto da Empresa (Chef Nery): fundamenta o contexto da empresa contendo;
- Justificativa da Abordagem: tem como finalidade explicar o motivo da adoção da abordagem;
- Processo de Engenharia de Requisitos: tem como finalidade apresentar o Processo de Engenharia de Requisitos de Software para a Fábrica de Massas;
- Elicitação de Requisitos: apresenta como serão feitas as elicitações dos requisitos;
- Tópicos de Gerenciamento de Requisitos: tem como finalidade mostrar como serão aplicadas as praticas de rastreabilidade de requisitos e os atributos que serão utilizados para os mesmos;
- Planejamento do Projeto;
- Ferramentas de Gestão de Requisitos: apresenta ferramentas de gerencia de requisitos e as analisa de acordo com características;
- Considerações Finais;
- Referências;





# Parte I

## Justificativa da Abordagem



## 2 Justificativa da Abordagem

A partir da década de 1970, a mudança tecnológica possibilitou um crescimento exorbitante dos recursos de hardware e permitiram que produtos com tecnologias mais avançadas fossem criadas. Entretanto, as técnicas utilizadas no desenvolvimento de software não acompanharam o crescimento dos recursos de hardware, este fenômeno ficou conhecido com a Crise de Software. Neste âmbito, Dijkstra destaca:

*A maior causa da crise do software é que as máquinas tornaram-se várias ordens de magnitude mais potentes! Em termos diretos, enquanto não havia máquinas, programar não era um problema; quando tivemos computadores fracos, isso se tornou um problema pequeno e agora que temos computadores gigantescos, programar tornou-se um problema gigantesco.*

Não obstante, ao longo dos anos, notou-se o aumento de metodologias de produção de software com objetivo de aumentar a qualidade e eficiência neste processo. A IEEE define metodologias de desenvolvimento de software como uma abordagem sistemática obtida na etapa de desenvolvimento, no qual, o papel do engenheiro é assegurar que sejam realizadas as melhores escolhas naquele determinado contexto. Vale ressaltar, que estas escolhas têm implicações diretas no sucesso ou fracasso de um projeto.

Na etapa de seleção de metodologia foi utilizada o fluxograma proposto pela empresa de consultoria Fortezza Consulting [N]. Apresentado abaixo:

Desta forma, foram considerada as seguintes perguntas:

1. *“Does the scope include software enablement of business processes?”*
2. *“Does the scope include software enablement of business processes?”*
3. *“Is the scope lacking in specificity, and unlikely to remain stable?”*
4. *“Is the customer willing and able to offer flexibility on scope?”*

Com base nas respostas das perguntas acima, foi identificada que a metodologia ágil se adaptaria melhor ao nosso contexto.

Além disso, uma série de fatores que justificaram o emprego da metodologia ágil, os quais estão descritos abaixo:

1. Flexibilidade a mudanças;
2. Comunicação com o cliente;
3. Planejamento;
4. Equipe de desenvolvedores;
5. Tamanho do Projeto;

### Flexibilidade a mudanças

O cliente, dono da fábrica de massas “Chef Nery”, ainda não alcançou uma idéia final concreta sobre o sistema que deseja adquirir. Os requisitos, embora já bem direcionados, permanecem instáveis no que diz respeito à definição de utilidades – ainda não é muito certo tudo o que será essencial ou apenas “bom ter” no projeto. Diante tal contexto, é natural que também o cliente esteja predisposto a flexibilizar o escopo da aplicação, permitindo que se assentem gradualmente as diretrizes definitivas do sistema.

Esta característica é favorável ao uso de metodologias ágeis, que preveem mudanças constantes no escopo e espera flexibilidade dos cliente envolvido [3].

### Comunicação com o cliente

O manifesto ágil, criado por em 1997, propõe uma série de princípios. Dentre eles, destaca-se a interação entre indivíduos como sendo mais relevante que os processos e ferramentas [3]. Assim, é necessário que os indivíduos tenham uma comunicação bastante próxima com o cliente e que as interações com o mesmo possam propiciar mudanças e evolução constantes, possibilitando a obtenção de requisitos voláteis.

O pequeno tamanho da equipe, que é formada por apenas 4 estudantes de Engenharia de Software e o cliente, somado ao fato de o cliente

estar apto e disposto a participar de reuniões e atender as necessidades da equipe de desenvolvimento, possibilita a comunicação próxima da qual as metodologias ágeis se beneficiam. Para tanto, foram adotadas algumas ferramentas de comunicação como o WhatsApp e Hangouts (para situações de impossibilidade de encontros físicos ou necessidade de reuniões emergenciais).

## Planejamento

Em metodologias ágeis, usualmente, as equipes economizam um tempo significativo com o planejamento, visto que as interações que ocorrem junto ao cliente fornecem feedbacks constantes. A título de comparação, em metodologias tradicionais aproximadamente 20 por cento do tempo é gasto na etapas de planejamento e replanejamento [4].

Não obstante, o cliente em nosso contexto não exige documentação expressiva, pelo contrário valoriza as interações provenientes da metodologia ágil. Desta forma, neste contexto específico é mais interessante adoção da metodologia ágil, pois esta possibilita uma economia de recursos e tempo.

## Equipe de desenvolvedores

A equipe de projetistas e desenvolvedores (que, na verdade, é a mesma equipe) já possui melhor familiaridade com metodologias ágeis, compreendendo relativamente bem suas etapas e rituais.

Somado a isto, há o tamanho diminuto da equipe, que consta 4 integrantes e o cliente, o que possibilita e também facilita uma comunicação intensa e frequente dos integrantes entre si e com o cliente.

Estas características tendem a uma menor dependência de formalidades em documentações e grande dinamicidade na gerência da equipe – ambos fatores que favorecem o uso de metodologias ágeis [5].

## Tamanho do Projeto

De acordo com o manifesto ágil, é necessário que os indivíduos tenham uma comunicação bastante próxima com o cliente (BECK et al, 2001). Não obstante, em projetos menores a comunicação é privilegiada visto a maior interação entre os integrantes do projeto.

Além disso, deve-se destacar que em projetos mais complexos, usualmente, é preferível a utilização de metodologias híbridas ou tradicionais, visto que estas detalham e documentam de forma mais precisa os processos e subprocessos.

Em projetos menores este formalismo é menos relevante, pois existe uma comunicação e um feedback maior entre os envolvidos.

## Scaled Agile Framework(SAFe)

O SAFe tem como objetivo sintetizar certos conceitos do corpo do conhecimento e as lições aprendidas de centenas de desenvolvimentos de projetos dentro de um único framework, a partir dos seguintes valores:

1. Alinhamento (Alignment);
2. Transparência (Transparency);
3. Execução do Programa (Program execution);
4. Qualidade(Built-in quality);

Assim, este framework escalável tem como objetivo prover práticas providas de metodologias ágeis, tais como SCRUM e XP em projetos. Além disso, este framework provém artefatos, documentos e atividades e pode ser dividido em três níveis: Portfólio, Programa e Time.

A partir destas características, foi escolhido o SAFe 4.0 ao nosso con-

texto. Entretanto, foram realizadas algumas adaptações para serem adaptadas ao nosso contexto. Vale ressaltar que o SAFe 4.0 propõe o Value Stream, o qual não foi considerado na elaboração do nosso processo, visto o tamanho e a simplicidade do projeto





## Parte II

### Processo de Engenharia de Requisitos



## 3 Processo de Engenharia de Requisitos

### Processo de Engenharia de Requisitos

#### Níveis do Processo

A seguir serão apresentados os níveis de abstração do processo de Engenharia de Requisitos. Ressalta-se o fato que este processo foi baseado no SAFe 4.0. A seguir serão apresentados estes níveis com maior detalhamento

#### Nível de Portfólio

O nível de Portfólio é o nível de abstração mais alto do SAFe. Nesse nível são providos as estruturas necessárias para o desenvolvimento do projeto. Nele são desenvolvidos: Enables de Épicos; Temas de Investimentos; e os Épicos.

#### Nível de Programa

O nível de Programa é o intermédio entre o nível de programa e o do Time. Desta forma, neste nível os times de desenvolvimento e outros recursos são aplicados na tarefa de desenvolvimento do projeto. Nesta etapa ocorre o refinamento dos Épicos que se transformam em Features e o planejamento do Program Increment (PI).

#### Nível de Time

Este nível se assemelha bastante ao SCRUM, portanto existem algumas atividades, tais como : Sprint planning, sprint review, sprint retrospective. Nesta etapa, as Features são fragmentadas em Histórias do Usuário. Esta etapa está relacionada com o desenvolvimento dos PI's.

### Papéis do Processo

A seguir serão apresentados os papéis adotados para o processo de Engenharia de Requisitos da Fábrica de massas ("Chef Nery"). Ressalta-se o fato que este processo foi baseado no SAFe 4.0. A seguir serão apresentados os papéis do processo.

### Papéis do Nível de Portfólio

#### Gerente de Portfólio

O Gerente de Portfólio tem como responsabilidade elaborar e comunicar temas estratégicos que implicam nas estratégias da empresa. Ressalta-se também o fato de ser responsável pela elaboração e priorização dos Épicos.

Em nosso contexto, o integrante Eduardo Gomes será o responsável por este papel.

#### Arquiteto do Sistema

O Arquiteto de Sistemas tem a responsabilidade de manter as soluções da empresa de forma holística e iniciativas de desenvolvimento. Além disso, deve fornecer caminhos arquiteturais da empresa que possibilite o suporte aos Enables de Épicos.

Neste contexto, o integrante Miguel Pimentel será o responsável por

este papel.

## Papéis do Nível de Programa

### Gerente do Produto

O gerente do Produto (Product Manager) tem como responsabilidade: compreender e entender as necessidades do cliente; validar as soluções desenvolvidas para a necessidade do cliente; compreender e auxiliar as atividades do nível de Portfólio. Priorizar o fluxo de trabalho do PI; desenvolver Visão; e gerenciar Release.

Em nosso contexto, o papel de gerente do produto será empenhado pelo integrante Miguel Nery.

## Papéis do Nível de Programa

### Scrum Master

O Scrum Master é responsável por assegurar as práticas e princípios do SCRUM e XP. Este papel também deve manter o foco do time, e assim, facilitar o progresso do time.

Em nosso contexto, O Scrum Master será um papel rotativo entre todos os integrantes do time.

### Product Owner

O Product Owner (PO) é o membro do time responsável pela definição das histórias e priorização do Backlog tanto como simplificar as prioridades do programa.

O PO também pode realizar algumas tarefas conhecidas, tais como:

desenvolvimento e participação no PI planning; definir metas da sprint; definir prioridades a partir do backlog; execução do programa; manter o programa; trabalhar em harmonia com o Product Manager; participar do planejamento e validação da sprint.

Em nosso contexto, PO será empenhado por Pedro Nery (Dono da fábrica de massas “Chef Nery”).

### Desenvolvedores (Time)

O time também conhecido como Equipe de Desenvolvedores está relacionado com o terceiro nível de abstração. Além disso, é responsável pela entrega de uma versão usável de cada incremento provido de um PI.

Em nosso contexto, todos os integrantes da equipe realizaram o papel de desenvolvedores.

### Artefatos do Processo

A seguir serão apresentados os artefatos oriundos do processo de obtenção de Requisitos da Fábrica de massas ("Chef Nery"). Ressalta-se o fato que este processo foi baseado no SAFe 4.0. A seguir serão apresentados os artefatos como uma breve descrição destes.

### Artefatos em Nível de Portfólio

#### Temas de Investimento

Temas estratégico é artefato responsável por todas as questões de negócio do projeto. Este artefato é relacionado com o nível de abstração Portfólio.

## Backlog de Épicos

O Backlog de Épicos é um artefato que engloba os épicos que foram levantados juntos a cliente e serão produzidos. Vale ressaltar que eles possuem uma priorização.

## Artefatos em Nível de Programa

### Visão

A artefato de visão descreve uma noção das soluções a serem desenvolvidas, refletindo as necessidades do cliente e stakeholders, como também as features e soluções propostas a estas necessidades.

### Backlog de Features

Apresenta o conjunto de features que serão implementadas para o sistema, estas devem estar pontuadas de acordo com a sua dificuldade técnica. Pode-se utilizar ferramentas como o planning poker para a avaliação da dificuldade técnica das features levantadas.

### Roadmap

Apresenta os entregáveis próximos a partir de uma linha do tempo, em outras palavras, apresenta as features organizadas de acordo com a priorização definida previamente.

### Especificação Suplementar

Este artefato apresenta todas as definições necessárias do negócio não incluídas na definição de Requisitos funcionais, no caso, Backlog de Features e Backlog do Time.

### Artefatos em Nível de Time

#### Backlog do Time

Representa um conjunto de histórias priorizada de acordo com os interesses do cliente. Elas consistem da fragmentação e refinamento das features definidas no nível do programa. Além disso, o Backlog do time também pode ser entendido como o conjunto de todas as coisas que o time deve fazer.

#### Sprint Backlog

Consiste em todas as histórias de usuário que devem ser desenvolvidas durante uma sprint. Vale ressaltar que estas histórias são oriundas do Backlog do Time.

### Atividades do Processo

As atividades estão dispostas a seguir de acordo com o quadro apresentado abaixo. Não obstante, cada atividade se encontra no subtópico referente ao seu nível de abstração.

No quadro abaixo, o identificador foi criado para identificar a atividade de acordo com o seu nível de abstração. Desta forma, os identificadores estão representados da seguinte maneira:



- Atividades que comecem com as iniciais PO, referem-se ao nível de abstração Portfólio e o número da atividade neste nível. Exemplo: PO01 (Atividade 1 do nível de abstração Portfólio);
- Atividades que comecem com as iniciais PR, referem-se ao nível de abstração Programa e o número da atividade neste nível. Exemplo: PR01(Atividade 1 do nível de abstração Programa);
- Atividades que comecem com as iniciais T, referem-se ao nível de abstração Time e o número da atividade neste nível. Exemplo: T03(Atividade 3 do nível de abstração Time);

Além disso, para atividades que não são necessárias entradas ou possuem não apresentam nenhuma saída será identificado pela sigla N/A (Não se aplica).



## 4 Modelo de Qualidade de Processo

### 4.0.1 Modelo de Qualidade de processo

Práticas implementadas no projeto:

*\*Para detalhes sobre as atividades mencionadas, ver tabelas na seção Atividades.*

#### 1. Nível G - Parcialmente Gerenciado

- Gerência de Requisitos (GRE)

*Propósito: gerenciar os requisitos do produto e dos componentes do produto do projeto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto.*

Resultados esperados:

- **GRE 1. (implementado)** O entendimento dos requisitos é obtido junto aos fornecedores de requisitos.

Implementado na seção de Portfólio do processo de Requisitos, através das atividades Compreender o contexto e os objetivos do cliente (PO01), Definir Épicas (PO02) e Priorizar Épicas (PO05). Também implementado na seção de Programa do processo de Requisitos, através das atividades Definir Features (PR03) e Priorizar Features (PR06). Também implementado na seção de Time do processo de Requisitos, através das atividades Definir Histórias do Usuário (T01) e Priorizar Histórias do Usuário (T03).

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

Implementado na seção de Portfólio do processo de Requisitos, através das atividades Definir épicos (PO02), Definir enables para os épicos (PO3) e Priorizar épicos (PO04). Também implementado na seção de Programa do processo de Requisitos, através das atividades Elaborar Visão (PR01), Definir features (PR03), Definir requisitos não-funcionais (PR04), Definir enables para features (PR05) e Priorizar features (PR06). Também implementado na seção de Time do processo de Requisitos, através da atividade Definir histórias de usuário (T01) e Priorizar histórias de usuário (T03).

- **GRE 3. A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;**
- **GRE 4. (implementado) Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;**

Implementado na seção de Portfólio do processo de Requisitos, através da atividade Validar Épicos (PO05). Como é possível ver no diagrama do processo, a continuação do processo após a PO05, que essencialmente é uma atividade de revisão, é condicionada por seu resultado. Caso a validação seja positiva, prossegue-se com o fluxo normal do processo; caso contrário, o processo retorna à estágios anteriores, para correção. Também implementado na seção de Programa do processo de Requisitos, através da atividade Validar features (PR08). A continuação do processo após a atividade PR08 é condicionada de forma análoga a da atividade PO05, citada acima, em função de seu resultado.

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

Implementado nas seções de Programa e Time do processo de Requisitos, através da atividade Gerenciar requisitos (PR02 e T02).

- **Nível D - Largamente definido:**

- Desenvolvimento de Requisitos (DRE)

Propósito: *Propósito: definir os requisitos do cliente, do produto e dos componentes do produto.*

Resultados esperados:

- **DRE 1. As necessidades, expectativas e restrições do cliente, tanto do produto quanto de suas interfaces, são identificadas;**

Implementado na seção de Portfólio do processo de Requisitos, através da atividade Compreender o contexto e os objetivos do cliente (PO01), Definir Épicos (PO02) e Priorizar Épicos (PO04). Também implementado na seção de Programa do processo de Requisitos, através das atividades Definir Features (PR03) e Priorizar Features (PR06). Também implementado na seção de Time do processo de Requisitos, através das atividades Definir Histórias do Usuário (T01) e Priorizar Histórias do Usuário (T03).

- **DRE 2. Um conjunto definido de requisitos do cliente é especificado e priorizado a partir das necessidades, expectativas e restrições identificadas;**

Implementado na seção de Portfólio do processo de Requisitos, através da atividade Priorizar Épicos (PO05). Também implementado na seção de Programa do processo de Requisitos, através da atividade Priorizar features (PR06). Também implementado na seção de Time do processo de Requisitos, através das atividades Definir Histórias do Usuário (T01) e Priorizar Histórias do Usuário (T03).

- **DRE 3. Um conjunto de requisitos funcionais e não-funcionais, do produto e dos componentes do produto que descrevem**

**a solução do problema a ser resolvido, é definido e mantido a partir dos requisitos do cliente;**

As atividades citadas no DRE 2, conjuntamente com a atividade Elaborar Requisitos não-funcionais (PR04) são responsáveis pela geração dos conjuntos de requisitos funcionais e não-funcionais, e a atividade Gerenciar requisitos (PR02 e T02) as mantém.

- **DRE 4. Os requisitos funcionais e não-funcionais de cada componente do produto são refinados, elaborados e alocados;**

Implementado na seção de Portfólio do processo de Requisitos, através da atividade Priorizar Épicos (PO04). Também implementado na seção de Programa do processo de Requisitos, através das atividades Elaborar Visão (PR01), Priorizar features (PR06), Elaborar Roadmap (PR07) e Elaborar PI (PR09). Também implementado na seção de Time do processo de Requisitos, através das atividades Priorizar Histórias do Usuário (T03) e Sprint Planning (T04).

- **DRE 5. Interfaces internas e externas do produto e de cada componente do produto são definidas;**

Interfaces do produto são definidas nas atividades mencionadas em DRE1. Interfaces das componentes do produto não são definidas; o pequeno tamanho do sistema demanda pouca necessidade de tal atividade e portanto ela será ignorada.

- **DRE 6. Conceitos operacionais e cenários são desenvolvidos;**

Conceitos operacionais e cenários não serão desenvolvidos. O nível de formalidade do projeto demanda pouca necessidade de tais atividades e portanto elas serão ignoradas.

- **DRE 7. Os requisitos são analisados, usando critérios definidos, para balancear as necessidades dos interessados com as restrições existentes;**

Não foram definidos critérios para tal atividade.

- **DRE 8. Os requisitos são validados.**

Implementado na seção de Portfólio do processo de Requisitos, através da atividade Validar Épicas (PO05). Também implementado na seção de Programa do processo de Requisitos, através das atividades Validar Features (PR08).

fonte: SOFTEX, Guia Geral MPS de Software.





## 5 Elicitação de Requisitos

Uma Elicitação de requisitos tem por característica achar o que realmente o usuário precisa para uma futura implementação de um software, é um processo que define o que o software deve ter como funcionalidade. Elicitar também pode ser descrito como descobrir ou obter informação sobre algo. O sucesso de um sistema de informação depende da qualidade da definição dos requisitos [10]. A isso se deve a grande importância de fazer um bom levantamento de requisitos.

Diversos projetos de software possuem a característica de terem falhado por problemas de Elicitação de Requisitos [8] [9]. Pode ser entendido que muitas vezes, os requisitos são mal interpretados ou incompletos. Com uma elicitação mal feita estariam comprometidos tanto a documentação quanto a especificação dos requisitos, provocando a inviabilidade de todo processo de Engenharia de Requisitos.

O processo de Engenharia de requisitos é a primeira etapa dentro de todo o processo de Engenharia de Requisitos [11]. Nela, a principal preocupação é em como levantar os reais requisitos do sistema. Para isso, existem diversas maneiras de levantar requisitos.

### 5.1 Técnicas de Elicitação de Requisitos

Foram estabelecidos critérios para a seleção das técnicas de elicitação de requisitos usadas. Estes foram:

- Disponibilidade da Equipe;
- Disponibilidade do Cliente;
- Compatibilidade das Técnicas com a abordagem.

A partir desses critérios serão/foram utilizadas as seguintes técnicas:

### 5.1.1 Observação

Essa técnica tem por característica possibilitar um contato pessoal estreito do pesquisador com o fenômeno pesquisado [12], no caso do presente projeto, um dos integrantes tem ligação parentesca com o cliente, isso facilita muito tanto a comunicação quanto o entendimento do trabalho para todo o grupo.

### 5.1.2 Entrevista

Essa é uma técnica que tem como base o engenheiro ou analista discutir o sistema com diferentes usuários, a partir disso elabora um entendimento dos requisitos [12]. No caso foi feita uma entrevista aberta, em que houve uma conversa para determinar o que o cliente necessita a partir de seus problemas.

### 5.1.3 Protótipo

É um sistema sem funcionalidades inteligentes. O protótipo cria uma interface para o cliente validar o software que será implementado.

### 5.1.4 Workshop

É uma técnica tem um objetivo pré determinado onde cada indivíduo tem direito a fala e são discutidos os requisitos de acordo com esse objetivo. Em um Workshop há um indivíduo que é responsável por conduzir a reunião e gerar discussões entre os envolvidos.

## 6 Gerenciamento de Requisitos

Gerenciar Requisitos é um processo associado à qualidade do desenvolvimento de software. Ela está tem a característica de ser um processo de entender e controlar diversas mudanças que ocorrem nos requisitos, por diversos motivos, como por exemplo mudanças no ambiente do sistema ou nos objetivos de uma organização [14], ele tem como principal objetivo a aquisição de conhecimentos das regras de negócios e verificação do que o cliente necessita para então obter uma boa especificação dos requisitos de software [13]. Para isso, neste tópico, foram estabelecidos alguns temas como a rastreabilidade dos requisitos e também os atributos dos requisitos que serão gerenciados.

### 6.1 Rastreabilidade de Requisitos

A rastreabilidade dos requisitos está diretamente ligada para referenciar um grupo coletivo de requisitos baseada em seus relacionamentos [15], ela estabelece formas de analisar o quanto mudanças afetaram o sistema.

Os elementos para estabelecer os relacionamentos entre os artefatos de software e os requisitos são chamados de elos, estes são elementos que são necessários para estabelecer a Rastreabilidade [15] e a partir deles pode ser levada em consideração um aspecto fundamental para esse contexto: a habilidade de descrever a “vida” de um determinado requisito.

O conceito de Rastreabilidade também pode ser definido como a capacidade de descrever e seguir o ciclo de vida de um requisito em diferentes direções [16]. Com isso os mais diversos requisitos e seus elos com determinados artefatos pode-se criar uma teia de relacionamentos em que a rastreabilidade tem como característica acompanhar justamente esses relacionamentos.

Para este projeto foi escolhida a técnica vertical em que tem como ca-

racterística relacionar artefatos dependendo de modelos. Para isso a rastreabilidade vertical será feita para Temas de Investimento, Épicos e Features e Casos de Uso.

## 6.2 Atributos dos Requisitos

Os requisitos do projeto irão ter alguns atributos que irão auxiliar no acompanhamento e gestão dos mesmos. Os atributos serão:

**Data de criação do requisito:** quando o requisito foi criado na ferramenta;

**Início previsto:** previsão de quando o requisito será desenvolvido;

**Término Previsto:** previsão de quando irá terminar o desenvolvimento do requisito;

**Data de início efetivo:** data em que foi iniciado o desenvolvimento do requisito;

**Data de conclusão do requisito:** quando o requisito terminou de ser desenvolvido;

**Valor para o negócio:** será classificada a relevância para o projeto o requisito em Alta, Média e Baixa:

- Prioridade Alta: É um requisito fundamental para a solução, o não atendimento desse requisito não atende a necessidade do cliente.

- Prioridade Média: É um requisito importante, porém a nível de satisfação do cliente mas não algo que é indispensável para a solução.

- Prioridade Baixa: É um requisito que seria bom ter, mas não agrega valor a solução e pode ter seu desenvolvimento adiado.

**Status do Requisito:** verifica a condição atual do requisito:

- Temas de Investimento e Épicos: podem estar nos estados Novo, Em Progresso e Feito.

- Features e Casos de Uso: podem estar nos estados Aberto, Em Progresso, Em Teste e Feito.

**Esforço:** Será uma característica que irá indicar o quanto um requi-

sito toma de esforço para ser concluído. Para isso será utilizado o Planning Poker, em que os envolvidos devem pontuar o quanto toma de esforço um requisito até chegarem a um consenso.



## 7 Ferramentas de Gestão de Requisitos

Foram realizadas pesquisas comparativas entre as ferramentas para gerir os requisitos, podendo elas serem versões Web ou ainda em versão de Desktop. Foram escolhidas então: Jira, Innoslate e Rally Dev.

### 8.1 Critérios

Para esta análise, foram listadas 5 características julgadas importantes para a gestão de requisitos para o projeto, referentes à Usabilidade, Rastreabilidade, Gestão de Mudanças, Flexibilidade e Licença.

**Usabilidade:** Analisa se a usabilidade da ferramenta é de fato intuitiva ou não, o que pode vir a trazer contratempos para a equipe.

**Licença:** Analisa a licença da ferramenta, caso seja grátis, paga, se possui isenção para projetos educativos, etc.

**Rastreabilidade:** Analisa se é possível rastrear de forma eficaz a ferramenta.

**Gestão de Mudanças:** Analisa quais são as funcionalidades que ajudam na gestão de mudanças, análise de impacto, e escopo.

**Flexibilidade:** Analisa a flexibilidade de personalização da ferramenta ao contexto do projeto.

### 8.2 Pontuação

Para conseguir escolher com exatidão, foi atribuído uma pontuação a tais características com o intuito de esclarecer numericamente qual a ferramenta que nos auxiliaria. A pontuação foi ponderada de 0 à 5 sendo mais próximo de 5 muito pertinente às necessidades do projeto, e mais próximo de 0 muito divergente do fim das reais necessidades do projeto.

### 8.3 Resultados

Item/Ferramenta	Atlasian Jira	Innoslate	Rally Dev
Usabilidade	4	3	3
Licença	3	4	4
Rastreabilidade	1	5	4
Gestão de mudanças	1	4	4
Flexibilidade	2	3	5
TOTAL	11	19	20

### 8.4 Descrição das notas atribuídas

#### **Usabilidade:**

Atlassian Jira: Interface bonita; Nomes bem intuitivos; Plugins gráficos interessantes.

Innoslate: Interface satisfatória;

RallyDev: Interface satisfatória; Ícones intuitivos;

#### **Licença:**

Atlassian Jira: Gratuito por 30 dias;

Innoslate: Versão grátis, com restrições;

RallyDev: Versão grátis, com restrições;

#### **Rastreabilidade:**

Atlassian Jira : Hierarquia pouco intuitiva; não existe ferramenta visual para controlar importância de requisitos em relação aos demais.

Innoslate: Rastreabilidade bastante intuitiva; Geração automática de índice de qualidade de requisitos e numeração na criação de entidade.

RallyDev: Hierarquia com fácil localização; Opção de linkar requisitos filhos;



**Gestão de mudanças:**

Atlassian Jira: Não aparenta ter algum feedback de mudanças do próprio autor ou de outros autores;

Innoslate: Controle de versão eficaz; Notificações com últimas alterações com autor e data;

RallyDev: Controle de versão eficaz;

**Flexibilidade:**

Atlassian Jira: Não apresenta ter opção de personalização;

Innoslate: Opções de fixar e desafixar abas importantes para o projeto;

RallyDev: Totalmente flexível para modificações relevantes para o projeto com inúmeras possibilidades de plugins.

## 8.5 Escolha da ferramenta

Após uma profunda análise em cada uma das ferramentas estudadas, foi decidido que o Rally Dev será utilizada para o gerenciamento de requisitos. Suas características de personalização e controle de versão foram fundamentais para a criação do próprio modelo de rastreabilidade do projeto.



## 8 Referências

1. Structured programming Academic Press Ltd. London, UK, UK ©1972
2. INSTITUTE OF ELECTRICAL AND ELETRONICS ENGINEERS. Guide to the Software Engineering Body of Knowledge, Version 3. [S.l.], 2014. 34-38 p.
3. BECK, K. et al. Agile manifest. 2001. Disponível em: <<http://www.agilemanifesto.com>>
4. BOEHM, B.; TURNER, R. Balancing Agility and Discipline: A Guide for the Perplexed.
5. 2001-2012 Scott W. Ambler  
<<http://www.agilemodeling.com/essays/agileDocumentation.htm>>
6. Martins L. E. G. e Daltrini B. M., Utilização dos preceitos da Teoria da Atividade na Elicitação dos Requisitos de Software, SBES'1999, pp.
7. <<https://www.rallydev.com/>>, acesso em 26 de Setembro de 2016.
8. Boehm, B. Software Engineering Economics, Prentice-Hall, 1981.
9. Goguen, J. A. and C. Linde “Techniques for Requirements Elicitation”, Software Requirements Engineering, 2nd. Ed., IEEE CS Press, 1997, pp 110-122.
10. Portella Cristiano R.R; Técnicas de prototipação na especificação de requisitos e sua influência na qualidade do software. Dissertação de Mestrado, Instituto de Informática PUC-Campinas, Campinas, 1994.
11. THAYER, R.H. e DORFMAN, M.; “Introduction to Tutorial Requirements Engineering” in Software Requirements Engineering. IEEE-CS Press, Second Edition, 1997, p.p. 1-2.
12. BELGAMO, Anderson; MARTINS, Luiz Eduardo Galvão. Estudo Comparativo sobre as técnicas de Elicitação de Requisitos do Software. In: XX Congresso Brasileiro da Sociedade Brasileira de Computação (SBC), Curitiba–Paraná. 2000.
13. DE GRANDE, José Inácio; MARTINS, Luiz Eduardo Galvão. SIGE-

RAR: Uma Ferramenta para Gerenciamento de Requisitos. In: WER. 2006. p. 75-83.

14. SOMMERVILLE, I,. “Engenharia de Software”, 6ª edição. Pearson Education do Brasil, 2003.

15. GENVIGIR, Elias Canhadas. Um modelo para rastreabilidade de requisitos de software baseado em generalização de elos e atributos. São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 2009.

16. Gotel, O. and Finkelstein, A. (1997) “Extended requirements traceability: Results of an industrial case study”. In Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, Washington, DC, p. 169.

## Referências