

# Ferramenta de Gerência de Requisitos

---

Requisitos de Software

RAFAEL FAZZOLINO PINTO BARBOSA - 11/0136942  
THIAGO RAMIRES KAIRALA - 12/0042916  
EDUARDO BRASIL MARTINS - 11/0115104  
BRUNO CONTESSOTTO BRAGANÇA - 09/0107853

Brasília, DF - 2014

# Histórico de Alterações

# Sumário

1	Introdução . . . . .	1
1.1	Propósito . . . . .	1
1.2	Escopo . . . . .	1
1.3	Definições, acrônimos e abreviações . . . . .	1
1.4	Visão geral . . . . .	2
2	Processo de Engenharia de Requisitos . . . . .	2
3	Documento de visão . . . . .	3
3.1	Posicionando . . . . .	3
3.1.1	Oportunidade de Negócios . . . . .	3
3.1.2	Instrução do Problema . . . . .	3
3.1.3	Instrução de Posição do Produto . . . . .	3
3.2	Descrições da Parte Interessada e do Usuário . . . . .	3
3.2.1	Resumo da Parte Interessada e do Usuário . . . . .	3
3.2.2	Principais Problemas e Necessidades da Parte Interessada . . . . .	4
3.3	Visão Geral do Produto . . . . .	5
3.3.1	Perspectiva do Produto . . . . .	5
3.3.2	Resumo das Capacidades . . . . .	5
3.4	Recursos do Produto . . . . .	6
3.4.1	Recurso 1 . . . . .	6
3.4.2	Recurso 2 . . . . .	6
3.5	Restrições . . . . .	6
3.6	Faixas de Qualidade . . . . .	6
3.7	Precedência e Prioridade . . . . .	6
3.8	Outros Requisitos do Produto . . . . .	6
3.8.1	Padrões Aplicáveis . . . . .	6
3.8.2	Requisitos do Sistema . . . . .	6
3.8.3	Requisitos de Desempenho . . . . .	6
3.8.4	Requisitos Ambientais . . . . .	6
3.9	Requisitos de Documentação . . . . .	6
3.9.1	Notas sobre a liberação, arquivo Leia-me . . . . .	6
3.9.2	Ajuda On-line . . . . .	6
3.9.3	Guias de Instalação . . . . .	6
3.10	Atributos do Recurso . . . . .	6
3.10.1	Status . . . . .	6
3.10.2	Prioridade . . . . .	6
3.10.3	Arquitetura . . . . .	6
4	Documento de casos de uso . . . . .	6
4.1	Objetivo . . . . .	6
4.2	Identificação dos atores . . . . .	6
4.2.1	Ator-01 Visitante . . . . .	6
4.2.2	Ator-02 Desenvolvedor . . . . .	6
4.3	Identificação dos Casos de Uso . . . . .	6
4.3.1	UC-01 Autenticar Usuário . . . . .	6
4.4	Diagrama de casos de uso . . . . .	6
4.5	Detalhamento dos casos de uso . . . . .	6
4.5.1	Caso de Uso: UC-01x Buscar cruzeiros . . . . .	6
5	Equipe . . . . .	6
6	Anexos . . . . .	7

# Lista de Figuras

1	Fishbone de Problema . . . . .	4
---	--------------------------------	---

# Lista de Tabelas

1	Parte Interessada . . . . .	3
2	Framework de Problema . . . . .	4
3	Framework de Necessidades . . . . .	5

# 1 Introdução

O desenvolvimento de software perpassa inúmeras fases até que o mesmo seja concluído e entregue ao cliente, uma dessas fases, e provavelmente a mais importante é a fase em que devemos entender o problema do usuário, compreender exatamente o que o mesmo necessita e apresentá-lo uma solução. Nesta fase, negociações serão feitas, tanto sobre funcionalidades do sistema quanto custo do projeto, tempo para conclusão do mesmo e restrições de projeto.

O resultado desta fase é uma documentação robusta, principalmente ao utilizar metodologias tradicionais de desenvolvimento. Nesta documentação se encontram as funcionalidades do software, as características do mesmo e as restrições de projeto, podendo abranger todo o software ou apenas uma primeira etapa de desenvolvimento, como é feito em metodologias ágeis, estes itens são conhecidos como requisitos.

Com esta documentação em mãos, os desenvolvedores podem começar o desenvolvimento do sistema, porém muitos problemas surgem tanto na construção da documentação quanto na utilização da mesma para o desenvolvimento do software. A gerência, organização, classificação e rastreabilidade dos requisitos geram inúmeros problemas, já que o conteúdo é muito extenso para ser organizado facilmente de forma adequada. A partir deste problema, surge a necessidade da utilização de ferramentas que auxiliem na organização, classificação e rastreabilidade dos requisitos.

## 1.1 Propósito

Ao analisar este documento, todos os *stakeholders* deverão compreender todo o contexto de negócio, os objetivos e escopo do projeto, assim como, entender o problema que deverá ser resolvido, quais necessidades do cliente deverão ser analisadas e quais serão as funcionalidades do sistema, assim como todas as suas características.

## 1.2 Escopo

Este documento abrange todo o contexto do desenvolvimento de software voltado para a fase de requisitos, desde a elicitação à gerência de requisitos. Encontra-se neste documento, o problema de negócio do cliente, suas reais necessidades, as características das mesmas e todas as funcionalidades do sistema.

Dessa forma, a partir deste documento, pode-se obter conhecimento total sobre o projeto de desenvolvimento da ferramenta de gerência de requisitos, desde a metodologia utilizada para o desenvolvimento até a forma de implementação do sistema.

## 1.3 Definições, acrônimos e abreviações

- *Stakeholders:*

Todos as partes envolvidas no contexto do sistema, desde o cliente e seus funcionarios até a equipe de desenvolvimento do sistema. Todos os interessados na solução de software são considerados stakeholders do sistema.

- *Requisitos:*

Engloba tudo que o software deve possuir para solucionar o problema em questão, desde funcionalidades do sistema até características que o software deve possuir.

- *Requisitos Funcionais:*

São chamados de requisitos funcionais todos aqueles que apresentam as funcionalidades do sistema. [Sommerville et al. 2003].

- *Requisitos não Funcionais:*

São chamados requisitos não funcionais todos aqueles que apresentam as características do sistema, incluindo compatibilidade, o tempo de resposta ou qualquer outra exigência que não inclua funcionalidades. [Sommerville et al. 2003].

- *Engenharia de Requisitos:*

Engenharia de Requisitos é um conceito que engloba todo um contexto de desenvolvimento de software que envolve elicitação de requisitos, negociação, verificação e validação, e documentação e gerência de requisitos para o desenvolvimento de um sistema computacional. O uso da palavra *Engenharia* garante que técnicas sistematicas serão utilizadas para que os requisitos sejam completos, corretos e consistentes [Espindola, Majdenbaum e Audy 2004].

- *Fishbone:*

Consiste em uma técnica utilizada para o reconhecimento do problema macro do cliente. A utilização desta técnica garante uma facilidade maior para entender o problema de negócio.

- *Framework do problema:*

Consiste em uma técnica para organizar e auxiliar o entendimento do problema, apresentar os stakeholders afetados pelo problema, o impacto que o problema gera para o cliente e uma possível solução bem sucedida. A utilização do framework garante maior facilidade no entendimento do contexto do cliente.

- *Framework de Necessidades:*

Consiste em uma técnica para organizar uma tabela identificando Necessidade, Problema, Solução atual e Solução Proposta. A utilização do framework de necessidade garante um melhor entendimento da necessidade do cliente.

- *WorkShop*

Workshop é uma técnica no qual os participantes discutem um problema em comum onde são aplicadas técnicas que ajudam em uma melhor identificação das necessidades do cliente e ajudam a melhorar o rendimento das reuniões.

- *Brainstorming*

Brainstorming é uma técnica que consiste em uma dinâmica de grupo para recolher ideias a respeito de um determinado assunto e para a resolução de problemas.

- *Casos de Uso:*

Caso de uso define uma sequência de ações que produz um resultado de valor observável. Os casos de uso fornecem estrutura para expressar requisitos funcionais no contexto dos processos de negócio e de sistema.

- *Sprint:*

Representa o espaço de tempo no qual deverão ser realizadas atividades previamente estabelecidas para a resolução de um problema. [Beck 2000].

- *Release:*

São entregas de código funcional, as quais são feitas por etapa, entregando pequenas partes do software de tempos em tempos. [Beck 2000].

- *Product Owner (PO):*

É o responsável pela atividade de repassar o conhecimento de todo o contexto de negócio para a equipe de desenvolvimento. Muitas vezes, o PO pode ser o próprio cliente ou qualquer funcionário que tenha conhecimento do problema e faz o intermédio entre a equipe de desenvolvimento e o cliente. [Beck 2000]

- *Product Backlog:*

Representa a produção do trabalho executado durante o desenvolvimento.[Sanches, Luiz et al. 2010].

- *Sprint Backlog:*

Representa o trabalho a ser desenvolvido durante uma *sprint* com o objetivo de criar um produto apresentável para a equipe. O *backlog* da *sprint* deve ser produzido de forma incremental.

## 1.4 Visão geral

Basicamente, neste documento, encontra-se todo o registro do contexto de desenvolvimento da Ferramenta de Gerência de Requisitos. O mesmo é organizado de forma a buscar o melhor entendimento a partir de qualquer *stakeholder* do projeto, desde leigos até funcionários da área.

## 2 Processo de Engenharia de Requisitos

COLOCAR AQUI O PROCESSO DE ENGENHARIA DE REQUISITOS UTILIZADO.

### 3 Documento de visão

O documento de visão tem como objetivo definir uma visão geral do projeto, apresentar os problemas, os requisitos funcionais, não funcionais, atores entre outras informações que serão definidos com o cliente a fim de garantir que a equipe de desenvolvimento e o cliente estejam na maior sincronia possível [IBM 2014].

#### 3.1 Posicionando

##### 3.1.1 Oportunidade de Negócios

Atualmente as ferramentas no mercado possuem limitações, como de qualidade, falta de flexibilidade na gerência, ou até mesmo o fechamento do código, que pode ser considerado uma limitação devida a redução de mão de obra para manutenção e evolução.

##### 3.1.2 Instrução do Problema

A Engenharia de Requisitos possui diversas *rotas* possíveis para se seguir, como por exemplo a rota ágil, tradicional ou até mesmo uma mistura das duas.

Infelizmente cada ferramenta de gerência de requisitos é voltada para uma dessas possibilidades, tornando difícil a tarefa voltada para outras, gerando assim nos engenheiros de requisitos a necessidade de aprender a utilizar diversas ferramentas para poder organizar projetos com *rotas* diferentes.

A utilização de apenas uma ferramenta que abrangesse as duas metodologias e ainda uma mistura das duas resolveria todo problema de gerência de requisitos em projetos que não se adequam a uma metodologia específica perfeitamente.

##### 3.1.3 Instrução de Posição do Produto

Para os engenheiros de Engenharia de Requisitos, a Nome da ferramenta representará um avanço nas atividades de gerenciamento, pois os mesmos apenas precisarão aprender as funcionalidades de uma ferramenta, simplificando a mudança entre projetos que tomam rotas distintas.

#### 3.2 Descrições da Parte Interessada e do Usuário

Nesta sessão serão identificados e detalhados os interessados e usuários da Nome da ferramenta.

##### 3.2.1 Resumo da Parte Interessada e do Usuário

Para melhor entendimento das características e responsabilidades dos interessados, utilizou-se uma tabela que apresenta todos os interessados no sistema, a descrição do mesmo, suas responsabilidades e os critérios de sucesso da sua função na equipe. Com esta tabela, pode-se obter o entendimento necessário sobre o nível de conhecimento dos interessados e o quanto importante eles são para o sucesso do sistema.

Interessado	Descrição	Responsabilidade	Critérios de Sucesso
Analista de Requisitos	Funcionário da equipe de desenvolvimento do <i>software</i> com facilidade em comunicação, psicologia, sociologia, filosofia e mais áreas votadas para o humano. Seu conhecimento na área pode ser, dependendo da organização, baixo.	Pessoa responsável para realizar a elicitação dos requisitos junto do usuário. Deve os requisitos da forma adequada para garantir sucesso no desenvolvimento do <i>software</i> .	Requisitos corretamente elicitados e prontos para serem documentados.
Gerente de Requisitos	Conhecedor de todo o processo de desenvolvimento de <i>software</i> , presente durante o processo e com contato frequente com o cliente. Seu conhecimento deve ser alto.	Pessoa responsável por administrar os requisitos durante todo processo de desenvolvimento de <i>software</i> , garantindo o mínimo esforço em casos de mudança de requisitos.	Requisitos fixos durante todo o processo, ou em casos de mudança de requisitos, gerção mínima de esforço na mudança do desenvolvimento.

Tabela 1. Parte Interessada



### 3.2.2 Principais Problemas e Necessidades da Parte Interessada

O problema a ser atacado pela solução de software deve estar bastante claro entre todos os *stakeholders* para que o desenvolvimento da ferramenta não passe por nenhuma dificuldade quanto ao entendimento de onde focar para realizar a solução do problema.

Para o entendimento do problema principal, foi utilizada a técnica de *fishbone*. Segue o resultado da mesma na figura abaixo:

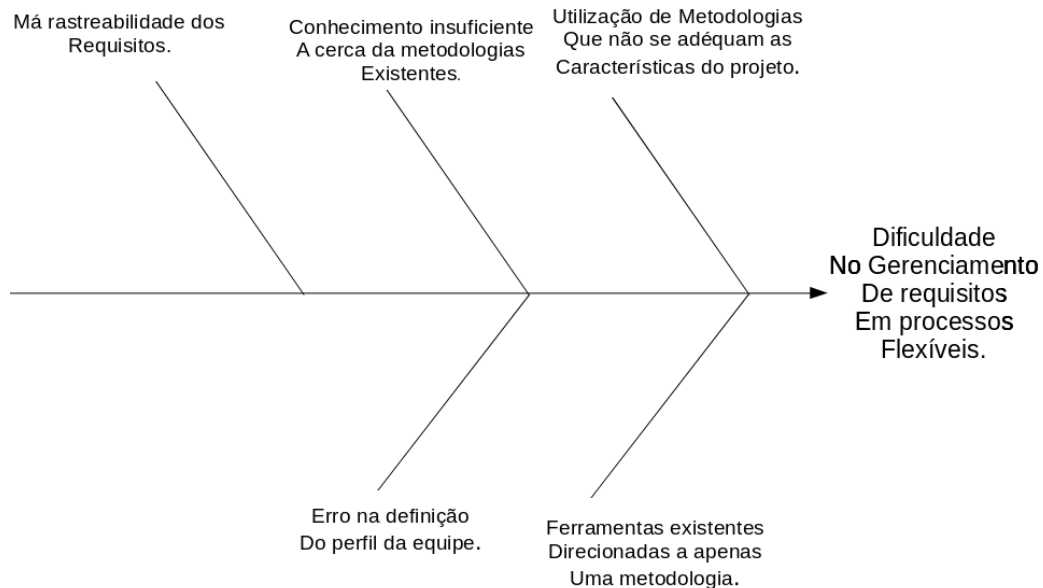


Figura 1. Fishbone de Problema

Para melhor entendimento do problema, utilizamos a técnica de *Framework de Problema*, na qual apresentamos o problema, os afetados, o impacto do problema e qual seria uma solução bem sucedida. Segue o *framework de Problema*:

<b>O Problema:</b>	Dificuldade no Gerenciamento de requisitos em processos flexíveis.
<b>Afeta:</b>	Todos os desenvolvedores de software que necessitam de uma flexibilidade maior na gerência de requisitos.
<b>Cujo impacto é:</b>	Software mal desenvolvidos, proporcionando maior possibilidade de erros.
<b>Uma solução bem sucedida seria:</b>	Utilização de uma ferramenta que faça gerência de requisitos de forma flexível, podendo utilizá-la em qualquer metodologia.

Tabela 2. Framework de Problema

Após o entendimento do problema, vê-se necessária a documentação das necessidades do cliente, para mesma, utilizou-se de uma técnica chama *framework de necessidades* na qual são apresentados todos os problemas, as necessidades, a solução atual e a solução proposta. Dessa forma, pode-se obter um entendimento mais organizado dos problemas e necessidades do cliente. Segue o *framework de necessidades*:

Necessidade	Problema	Solução Atual	Solução Proposta
Obter uma opinião da melhor metodologia a ser seguida de acordo com as características do projeto.	Utilização de metodologias que não se adequam as características do projeto.	A equipe de desenvolvimento escolhe a sua metodologia.	Criação de uma ferramenta que apresente a melhor opção de metodologia a ser seguida, de acordo com as características do projeto.
Obter a metodologia que melhor se adequa ao contexto, mesmo sem ter conhecimento profundo sobre a mesma.	conhecimento insuficiente a cerca das metodologias existentes.	Estudar as metodologias existentes.	Criação de uma ferramenta que faça a seleção da melhor metodologia sem que os envolvidos precisem ter conhecimento sobre a mesma.
Criação de uma rastreabilidade de fácil entendimento e bem organizada.	Má rastreabilidade dos requisitos.	Realizar a rastreabilidade a partir do conhecimento da equipe.	Utilização de uma ferramenta que documente os requisitos tomando cuidado com a rastreabilidade do mesmo, garantindo organização e fácil entendimento.
Necessidade de ferramentas flexíveis, que utilizam determinada metodologia de acordo com as características do projeto.	Ferramentas existentes direcionadas a apenas uma metodologia.	Utilização de mais de uma ferramenta. 4	Criação de uma ferramenta que utilize todas as metodologias, se adequando as características do projeto.
Obter o perfil da equipe já definido.	Erro na definição do perfil da equipe.	A própria equipe define seu perfil.	Criação de uma ferramenta que apresente o perfil da equipe a partir de <i>inputs</i> que representem algumas características da mesma.

**Tabela 3.** Framework de Necessidades

### 3.3 Visão Geral do Produto

Nesta seção, pode-se ter um entendimento geral de como será o produto final, quais serão suas características, como serão suas funcionalidades e etc.

#### 3.3.1 Perspectiva do Produto

O produto se encontrará em um contexto onde existem inúmeras ferramentas que possuem o objetivo principal do mesmo, porém, as ferramentas existentes possuem uma inflexibilidade muito grande relacionada a abordagem que será seguida durante o desenvolvimento de software. A qual o produto final deste desenvolvimento irá combater utilizando métodos de escolha da abordagem mais indicada para cada contexto de projeto.

A ferramenta pode ser autocontida, não necessitando do apoio de nenhum outro sistema, porém a utilização de ferramentas de modelagem de processos é bastante indicada para que a máxima organização do projeto seja alcançada.

#### 3.3.2 Resumo das Capacidades

O grande diferencial da ferramenta resultado deste desenvolvimento será a flexibilidade na abordagem que será seguida durante o gerenciamento de projetos de software. O sistema deverá indicar a melhor abordagem a ser seguida pela equipe de desenvolvimento, garantindo a otimização no desempenho durante o processo de desenvolvimento de software.

A ferramenta será capaz de disponibilizar a opção de modificar a abordagem indicada pela ferramenta, para que a equipe de desenvolvimento possa escolher a abordagem na qual os mesmos se sentem mais a vontade.

### **3 .4 Recursos do Produto**

#### **3 .4.1 Recurso 1**

#### **3 .4.2 Recurso 2**

### **3 .5 Restrições**

### **3 .6 Faixas de Qualidade**

### **3 .7 Precedência e Prioridade**

### **3 .8 Outros Requisitos do Produto**

#### **3 .8.1 Padrões Aplicáveis**

#### **3 .8.2 Requisitos do Sistema**

#### **3 .8.3 Requisitos de Desempenho**

#### **3 .8.4 Requisitos Ambientais**

### **3 .9 Requisitos de Documentação**

#### **3 .9.1 Notas sobre a liberação, arquivo Leia-me**

#### **3 .9.2 Ajuda On-line**

#### **3 .9.3 Guias de Instalação**

### **3 .10 Atributos do Recurso**

#### **3 .10.1 Status**

#### **3 .10.2 Prioridade**

#### **3 .10.3 Arquitetura**

## **4 Documento de casos de uso**

### **4 .1 Objetivo**

### **4 .2 Identificação dos atores**

#### **4 .2.1 Ator-01 Visitante**

#### **4 .2.2 Ator-02 Desenvolvedor**

### **4 .3 Identificação dos Casos de Uso**

#### **4 .3.1 UC-01 Autenticar Usuário**

### **4 .4 Diagrama de casos de uso**

### **4 .5 Detalhamento dos casos de uso**

#### **4 .5.1 Caso de Uso: UC-01x Buscar cruzeiros**

## **5 Equipe**

### **Alunos**

- Rafael Fazzolino Pinto Barbosa
- Thiago Ramires Kairala
- Eduardo Brasil Martins
- Bruno Contessotto Bragança

### **Professores**

- Mr. George Marsicano

## 6 Anexos

- **Entrevista realizada dia 16 de outubro de 2014**

**Cliente:** George Marsicano

**Questão 1:** Por que criar uma nova ferramenta de requisitos? As existentes nao te agradam?

**Resposta do cliente:**

Não me agradam? \*risos\*

A necessidade de criação ou não de uma ferramenta não vem da necessidade de ferramenta em si. Vocês em primeiro passo estão modelando um processo de requisitos. Depois será realizado uma avaliação para saber se existe ou não alguma ferramenta que implemente este processo definido, e, por ultimo, caso não exista nenhuma ferramenta, poderá ser desenvolvido um plug-in para alguma existente que seja possível tal adaptação, ou o desenvolvimento de uma nova ferramenta por completo.

Tudo dependerá a modelagem inicial para decidir o andamento e escopo do projeto.

# Referências Bibliográficas

[Beck 2000]BECK, K. Extreme programming explained: embrace change. [S.l.]: Addison-Wesley Professional, 2000.

[Espindola, Majdenbaum e Audy 2004]ESPINDOLA, R. S. de; MAJDENBAUM, A.; AUDY, J. L. N. Uma análise crítica dos desafios para engenharia de requisitos em manutenção de software. In: WER. [S.l.: s.n.], 2004. p. 226–238.

[IBM 2014]IBM. Rational Unified Process. out. 2014. Disponível em: <<http://pic.dhe.ibm.com/infocenter/rpcmpose/v2r0/in>

[Sanches, Luiz et al. 2010]SANCHES, F.; LUIZ, M. et al. Aplicação das abordagens scrum e xp em um processo de software. Sistemas de Informação & Gestão de Tecnologia., n. 3, 2010.

[Sommerville et al. 2003]SOMMERVILLE, I. et al. Engenharia de software. [S.l.]: Addison Wesley, 2003.