

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

Sigla	Significado
V	Versão
MF	Número de arquivos modificados.
AL	Número de linhas adicionadas.
DL	Número de linhas deletadas.

V	Autor	Data	Mensagem do Commit	MF	AL	DL
0	Rafael Fazzolino	2014-10-17	Criando a estrutura do documento	40	3598	0
1	Rafael Fazzolino	2014-10-17	Criação do contexto de negócio	5	35	31
2	RafaelFazzolino	2014-10-17	Atualizando detalhes da estrutura de documentação.	1	2	40
3	Thiagokairala	2014-10-21	Inseridos objetivos gerais e específicos e justificativa	10	174	139
4	Thiago	2014-10-22	Montado um template para o documento, e inserido parte de metodologia	21	385	1611
5	Rafael Fazzolino	2014-10-28	Voltando ao documento inicial	1	0	1
6	Eduardo Brasil	2014-10-28	Introdução Documento de Visão	2	89	143
7	Eduardo Brasil	2014-10-28	inserido primeira parte do visão	7	16	43
8	Rafael Fazzolino	2014-10-28	arrumando main	1	1	0
9	Rafael	2014-10-28	Criação das Definições, acrônimos e abreviações	6	62	2
10	Rafael Fazzolino	2014-10-28	Resolvendo conflitos de bibliografia	6	10	3
11	Rafael Fazzolino	2014-10-29	Atualizando visão e aplicando normas ao sumário	8	31	49
12	Rafael Fazzolino	2014-10-29	Criando item de problema e necessidade	4	27	4
13	Rafael Fazzolino	2014-10-29	Criando fishbone, framework de problema e resolvendo alguns erros	6	11	2
14	Rafael Fazzolino	2014-10-29	Criação do framework de necessidades e organização do documento	8	47	9
15	Bruno Contessotto	2014-10-29	Criando anexos e adicionando entrevista	3	13	9
16	Rafael Fazzolino	2014-10-30	Adicionando Tabela de Usuários e Visão geral do produto	6	61	37
17	Thiago	2014-10-31	Inserindo processo de desenvolvimento	9	39	3
18	Thiago Kairala	2014-11-02	Revisando e corrigindo ortografia do documento	7	156	108
19	Thiago Kairala	2014-11-02	Revisada e corrigida introdução	21	45	199
20	Thiago Kairala	2014-11-02	preparando documento para usar com latex-tools	11	18	1
21	Thiago	2014-11-03	revisando documento como um todo e alterando anexo	16	50	26
22	Eduardo	2014-11-03	iniciando restrições	10	4	26
23	Rafael Fazzolino	2014-11-03	Novo fishbone	12	23	2
24	Thiago Kairala	2014-11-03	Adicionada matriz de rastreabilidade	13	24	32

Sumário

1	Introdução	1
1.1	Propósito	1
1.2	Escopo	1
1.3	Definições, acrônimos e abreviações	1
2	Processo de Engenharia de Requisitos	3
3	Documento de visão	5
3.1	Posicionando	5
3.1.1	Oportunidade de Negócios	5
3.1.2	Instrução do Problema	5
3.1.3	Instrução de Posição do Produto	5
3.2	Matriz de rastreabilidade de requisitos	6
3.3	Descrições da Parte Interessada e do Usuário	6
3.3.1	Resumo da Parte Interessada e do Usuário	6
3.3.2	Principais Problemas e Necessidades da Parte Interessada	6
3.4	Visão Geral do Produto	8
3.4.1	Perspectiva do Produto	8
3.4.2	Resumo das Capacidades	8
3.5	Recursos do Produto	9
3.5.1	Recurso 1	9
3.5.2	Recurso 2	9
3.6	Restrições	9
3.7	Faixas de Qualidade	10
3.8	Precedência e Prioridade	10
3.9	Outros Requisitos do Produto	10
3.9.1	Padrões Aplicáveis	10
3.9.2	Requisitos do Sistema	10
3.9.3	Requisitos de Desempenho	10
3.9.4	Requisitos Ambientais	10
3.10	Requisitos de Documentação	10
3.10.1	Notas sobre a liberação, arquivo Leia-me	10
3.10.2	Ajuda On-line	10
3.10.3	Guias de Instalação	10
3.11	Atributos do Recurso	10
3.11.1	Status	10
3.11.2	Prioridade	10
3.11.3	Arquitetura	10
4	Documento de casos de uso	10
4.1	Objetivo	10
4.2	Identificação dos atores	10

4 .2.1	Ator-01 Visitante	10
4 .2.2	Ator-02 Desenvolvedor	10
4 .3	Identificação dos Casos de Uso	10
4 .3.1	UC-01 Autenticar Usuário	10
4 .4	Diagrama de casos de uso	10
4 .5	Detalhamento dos casos de uso	10
4 .5.1	Caso de Uso: UC-01x Buscar cruzeiros	10
5	Road Maps	10

Lista de Figuras

1	Modelagem Parte 1	3
2	Modelagem Parte 2	4
3	Modelagem Parte 3	5
4	Matriz de rastreabilidade	6
5	Fishbone de Problema	7

Lista de Tabelas

2	Parte Interessada	6
3	Framework de Problema	7
4	Framework de Necessidades	8

1 Introdução

O desenvolvimento de *software* passa por inúmeras fases até que seja concluído e entregue ao cliente, uma delas, e provavelmente a mais importante, é a Engenharia de Requisitos, onde devemos entender o problema do usuário, compreender suas necessidades e apresentá-lo a uma solução. Nesta fase, negociações serão feitas, tanto sobre funcionalidades do sistema quanto custos, tempo para conclusão e restrições de qualquer tipo.

O resultado desta fase é uma documentação robusta, principalmente ao utilizar metodologias tradicionais de desenvolvimento. Nesta documentação encontram-se as funcionalidades do *software*, suas características e restrições, podendo abranger todo o *software* ou apenas uma primeira etapa de desenvolvimento, como é feito em metodologias ágeis.

A tarefa de construir e manter a documentação necessária em um projeto de *software* possui diversos problemas relacionados a diversas áreas diferentes, como por exemplo a gerência, organização, classificação e rastreabilidade dos requisitos. Surge assim a necessidade da utilização de ferramentas que possam amenizar as dificuldades encontradas.

1.1 Propósito

Ao ler 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.

1.2 Escopo

Este documento abrange o contexto do desenvolvimento de *software* voltado para a Engenharia de Requisitos, desde a elicitação à gerência de requisitos, e tem como objetivo levar o entendimento do projeto a qualquer leitor, desde leigos até especialistas na área. Encontra-se neste documento, o problema de negócio do cliente, suas reais necessidades, características e funcionalidades do sistema que foram possíveis mapear.

Dessa forma, a partir deste documento, pode-se obter conhecimento total sobre o projeto de desenvolvimento da R.A.D.I.T., desde a metodologia utilizada até a forma de implementação do sistema.

1.3 Definições, acrônimos e abreviações

Durante o processo de elicitação e gerenciamento de requisitos é necessário que todos os envolvidos possam se comunicar sem que existam falhas de entendimento, para isso, foi desenvolvido um sumário contendo nomes que serão utilizados no processo, assim como suas definições.

- *Stakeholders*

Todas 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 [Sommerville et al. 2003].

- *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 macro problema do cliente. A utilização desta técnica garante uma facilidade maior para entender onde a solução deve atuar.

- *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.

2 Processo de Engenharia de Requisitos

Inicialmente, foi necessário entender o problema do qual iríamos tratar, traçar características e definir uma visão com o cliente para impedir problemas futuros, como, por exemplo, problemas de comunicação causados por ambiguidade ou coisas parecidas, estas informações estão esclarecidas no Documento de Visão, presente na sessão 3 deste documento.

Após a definição do Documento de Visão iniciou-se a parte de elicitações de requisitos, a modelagem desta está representada na imagem 1

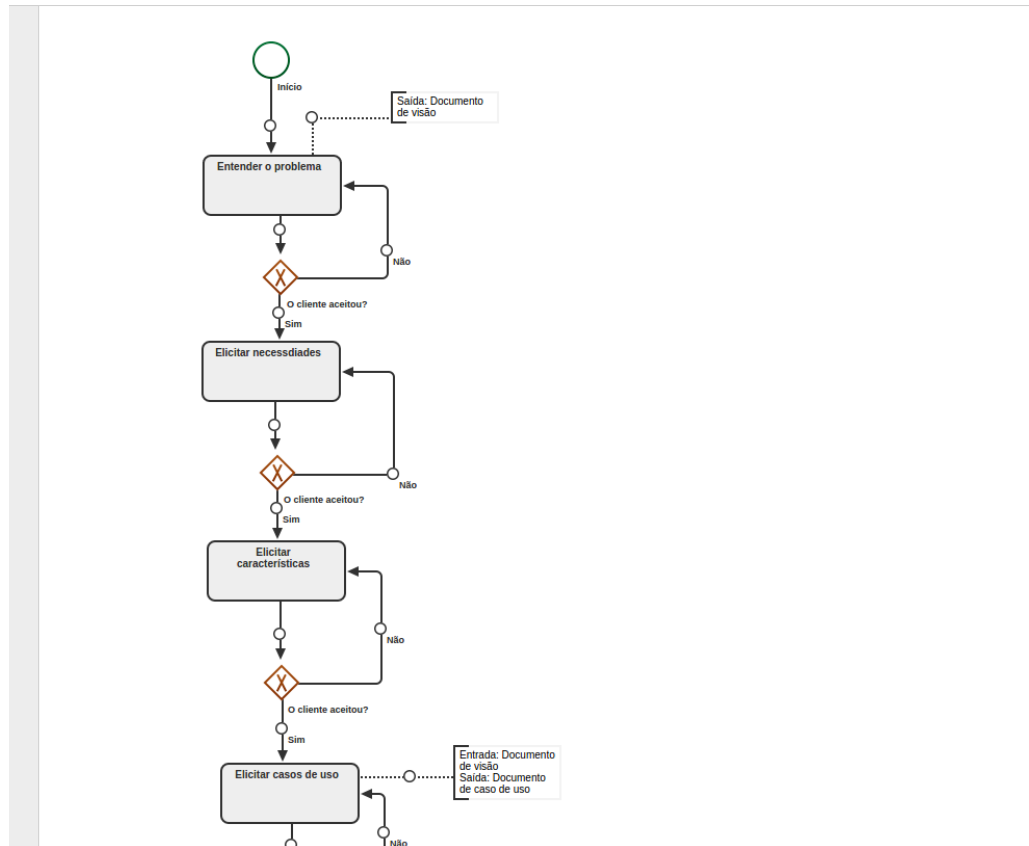


Figura 1. Modelagem Parte 1

Após os casos de uso do projeto definidos, vem a parte de definição de prioridades, criação dos *road maps*, assim como detalhamento dos casos de uso e implementação das funcionalidades com maior prioridade do projeto, a modelagem do mesmo esta presente na imagem 2.

O detalhamento dos casos de uso está presente na sessão 4 deste documento, assim como os roads maps se encontram na sessão 5 .

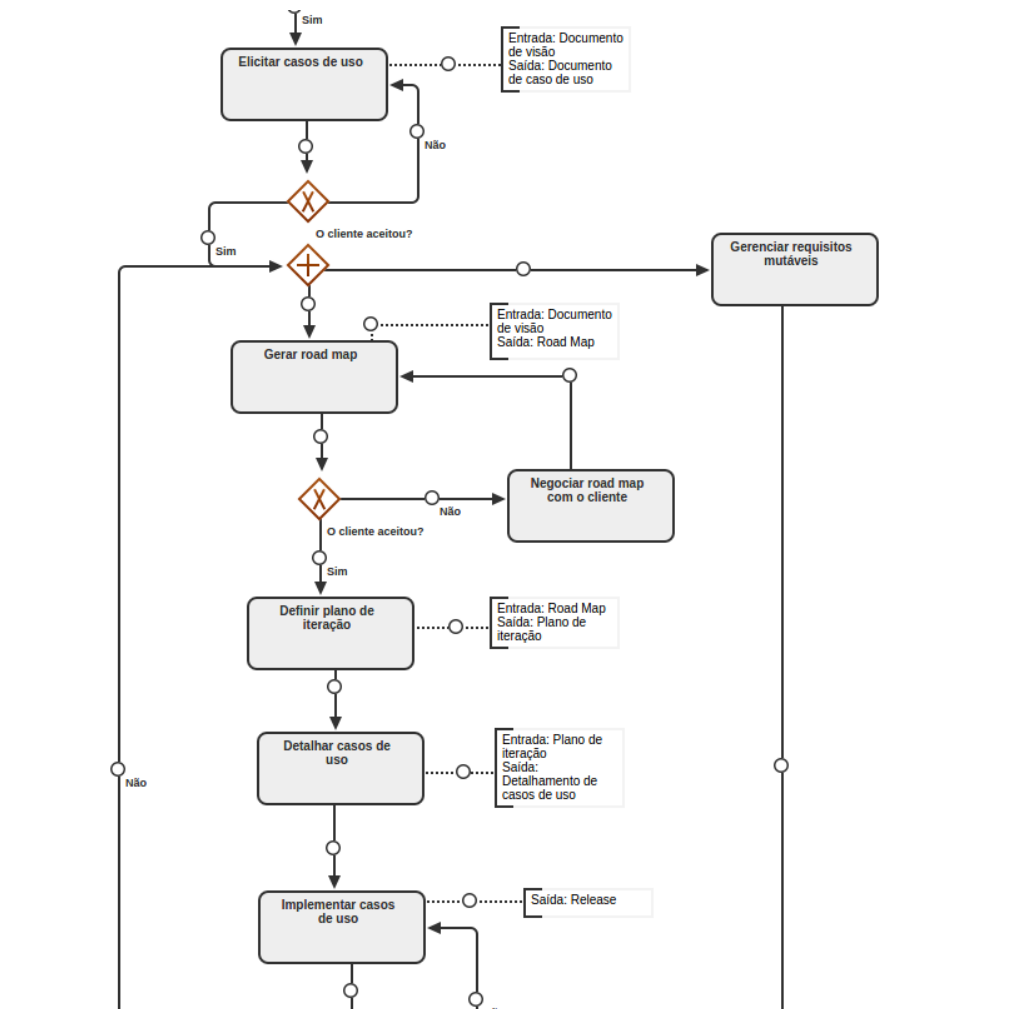


Figura 2. Modelagem Parte 2

Após a implementação dos casos de uso da iteração, caso o cliente aceite, segue-se para a próxima iteração ou final do projeto, dependendo se existe ou não outros casos de uso a serem implantados, caso haja, o processo volta para as atividades de gerar o road map e gerenciar requisitos mutáveis presentes na figura 2, assim como mostra a figura 3.

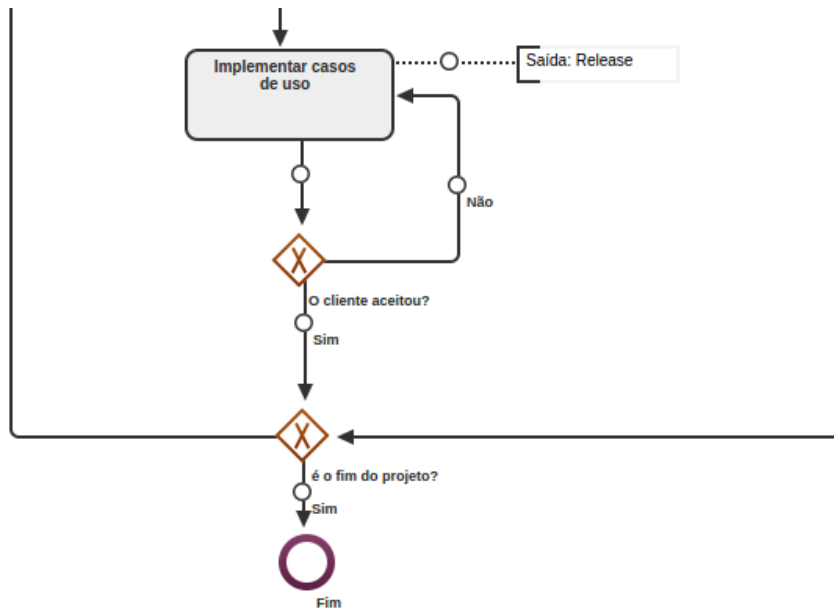


Figura 3. Modelagem Parte 3

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 R.A.D.I.T. 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 Matriz de rastreabilidade de requisitos

A matriz de rastreabilidade resume o modo como os requisitos do sistema serão organizados, e esta é ilustrada na imagem 4.

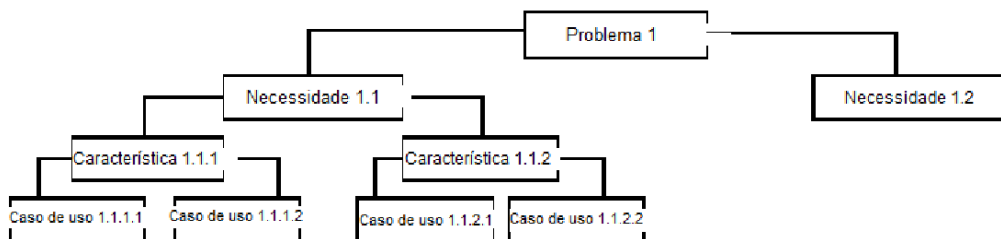


Figura 4. Matriz de rastreabilidade

3.3 Descrições da Parte Interessada e do Usuário

Nesta sessão serão identificados e detalhados os interessados e usuários da R.A.D.I.T..

3.3.1 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, suas descrições, responsabilidades e os critérios de sucesso de suas funções na equipe, ilustrada na tabela 2. Com esta tabela, pode-se obter o entendimento necessário sobre os interessados e o quão importante eles são para o sucesso do sistema.

Interessado	Descrição	Responsabilidade	Crítérios de Sucesso
Analista de Requisitos	Membro da equipe de desenvolvimento com facilidade em comunicação, psicologia, sociologia, filosofia e mais áreas que possam facilitar a relação com o cliente. Seu conhecimento na área pode ser, dependendo da organização, baixo.	Pessoa responsável por realizar a elicitação dos requisitos junto ao usuário. Deve elicitar os requisitos de forma adequada à 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 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 bem administrados para, no caso de mudanças nos requisitos, existir o menor impacto possível na equipe de desenvolvimento.

Tabela 2. Parte Interessada

3.3.2 Principais Problemas e Necessidades da Parte Interessada

O problema a ser resolvido pela R.A.D.I.T. deve estar bastante claro entre todos os *Stakeholders* para que o desenvolvimento passe pela menor quantidade possível de dificuldades quanto ao entendimento de onde focar esforços para desenvolver a solução do problema.

Para o mapeamento do problema principal e suas causas, foi utilizada a técnica de *fishbone*. Segue o resultado da mesma na figura 5

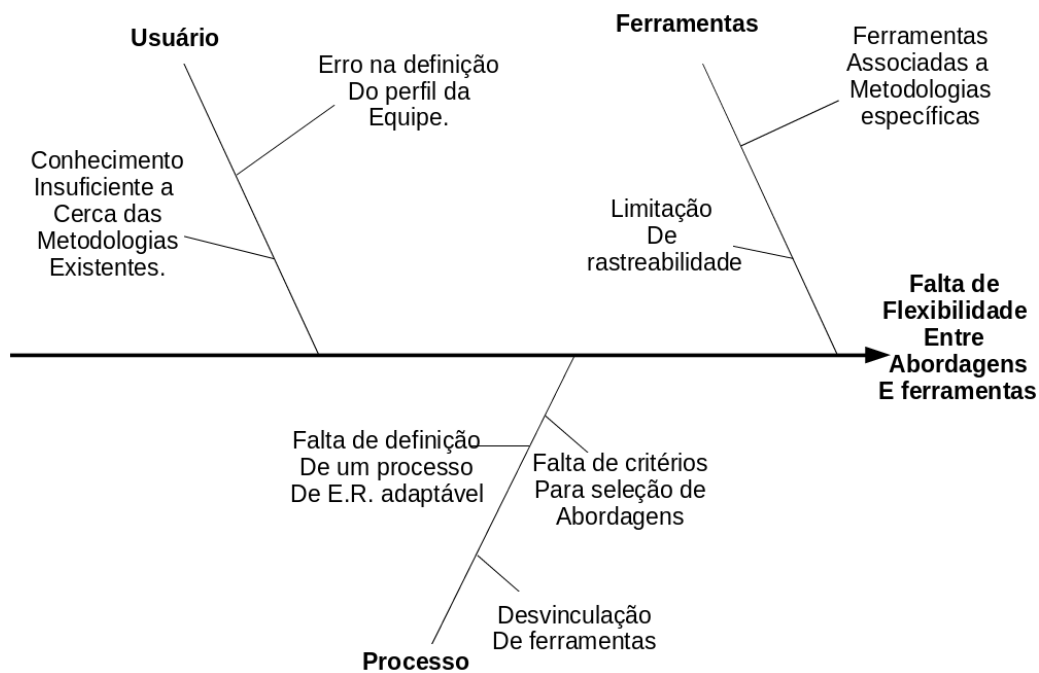


Figura 5. Fishbone de Problema

Para melhor entendimento do problema, utilizamos a técnica de *Framework de problema*, que consiste em criar uma tabela apresentando o problema, os afetados, o impacto do problema e qual seria uma solução bem sucedida, o Framework esta retratado na tabela 3.

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

Tabela 3. Framework de Problema

Após o entendimento do problema, vê-se necessária a documentação das necessidades do cliente. Utilizou-se de uma técnica chamada *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, de acordo com o retradado na tabela 4.

Necessidade	Problema	Solução Atual	Solução Proposta
Descobrir a metodologia ideal a ser utilizada no 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, equipe e cliente.
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.	Dificuldade em manter rastreabilidades de 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 em diferentes projetos.	Criação de uma ferramenta que se adequa a todas as metodologias, de acordo com 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 4. Framework de Necessidades

3.4 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.

3.4.1 Perspectiva do Produto

O produto se encontrará em um contexto onde existem inúmeras ferramentas com o mesmo propósito, porém, as ferramentas existentes são inflexíveis quando se trata da abordagem que será seguida durante o desenvolvimento de *software*. Esta falha será corrigida na R.A.D.I.T., que irá propor uma metodologia para cada projeto em particular de acordo com suas características.

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.

3.4.2 Resumo das Capacidades

O grande diferencial da R.A.D.I.T. será a flexibilização 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 do processo de desenvolvimento.

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 .5 Recursos do Produto

3 .5.1 Recurso 1

3 .5.2 Recurso 2

3 .6 Restrições

O sistema operacional poderá ser Windows e algumas destribuições de linux como Ubuntu, mint e open suse, o software também poderá ser executado pelos navegadores Google Chrome, Fire fox e só não será viavel sua utilização no Internet Explorer e Safari.

- 3 .7 Faixas de Qualidade
- 3 .8 Precedência e Prioridade
- 3 .9 Outros Requisitos do Produto
 - 3 .9.1 Padrões Aplicáveis
 - 3 .9.2 Requisitos do Sistema
 - 3 .9.3 Requisitos de Desempenho
 - 3 .9.4 Requisitos Ambientais
- 3 .10 Requisitos de Documentação
 - 3 .10.1 Notas sobre a liberação, arquivo Leia-me
 - 3 .10.2 Ajuda On-line
 - 3 .10.3 Guias de Instalação
- 3 .11 Atributos do Recurso
 - 3 .11.1 Status
 - 3 .11.2 Prioridade
 - 3 .11.3 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 Road Maps

AQUI DEVERÃO ENTRAR OS ROAD MAPS 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.

Anexos

•ENTREVISTAS

–Entrevista realizada dia 16 de outubro de 2014

Cliente: George Marsicano

Foi planejado pela equipe de desenvolvimento uma entrevista superficial com seis perguntas, porém, após a primeira pergunta respondida não foi mais necessário as outras perguntas, salvo a pergunta de número dois.

São elas:

1. Por que criar uma nova ferramenta de requisitos? As existentes não te agradam?

R: Não me agradam?

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 último, 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.

2. Se fosse possível resumir a sua necessidade em uma palavra, qual seria?

R: Abrangência.