Projeto Final da Disciplina de Requisitos de Software

Ferramenta de Gerência de Requisitos

Requisitos de Software

Bruno Contessotto Bragança - 09/0107853
Eduardo Brasil Martins - 11/0115104
Rafael Fazzolino Pinto Barbosa - 11/0136942
Thiago Ramires Kairala - 12/0042916

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, acrônomos 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
25	Eduardo	2014-11-03	retirando partes que não se aplicam	1	1	32
26	Bruno Bragança	2014-11-03	Anexos	12	74	10

Continua na página seguinte

V	Autor	Data	Mensagem do Commit	MF	AL	DL
27	Rafael Fazzolino	2014-11-03	Novo framework de problema e necessidades	4	35	20
28	Bruno Bragança	2014-11-03	Arrumando restrições	3	4	1
29	Thiago Kairala	2014-11-03	Inserindo atributos de recurso	6	122	13
30	Eduardo Brasil	2014-11-04	Inserido listagem dos recursos	3	26	17
31	Thiago kairala	2014-11-04	Inserido tabela de pontuação dos atributos	7	57	9
32	Thiago	2014-11-05	adicionando rastreabilidade dos casos de uso.	6	68	55
33	Thiago kairala	2014-11-04	Inserido tabela de pontuação dos atributos	2	140	7
34	Thiago kairala	2014-11-06	Corrigida rastreabilidade do documento	4	63	70
35	Thiago kairala	2014-11-06	Inserido pontuação de cada recurso, assim como gerado rank de todos eles	5	103	41
36	Thiago Kairala	2014-11-03	Inserindo atributos de recurso	1	0	0
37	Rafael Fazzolino	2014-11-06	Arrumando imagem do processo de E.R.	5	7	16
38	Thiago kairala	2014-11-06	Inserido roadmap	7	115	62
39	Bruno Bragança	2014-11-09	Arrumando roadmap	4	4	3
40	Rafael Fazzolino	2014-11-10	Adicionando tabela de rastreabilidade.	6	21	79
41	Eduardo Brasil	2014-11-10	Arrumando listagem de recursos	5	114	43
42	Thiago kairala	2014-11-11	Detalhado caso de uso da decisão da metodologia	3	57	3
43	Rafael Fazzolino	2014-11-11	Atualizando o RoadMap	5	27	24
44	Thiago kairala	2014-11-06	Inserido roadmap	4	18	2
45	Thiago kairala	2014-11-11	Inserindo requisitos não funcionais	2	25	6
46	Thiago kairala	2014-11-11	Inserindo requisitos não funcionais	1	2	2
47	Rafael Fazzolino	2014-11-11	Corrigindo alguns erros	4	5	2
48	Eduardo Brasil	2014-11-12	adicionando descrição de caso de uso	4	29	33
49	Eduardo brasil	2014-11-13	Iniciando questionario - descrição/objetivo	3	24	0
50	Eduardo Brasil	2014-11-13	Inserido descrições dos casos de uso	8	58	18
51	Thiago Kairala	2014-11-13	Terminando a breve descrição nos casos de uso	1	15	8
52	Thiago Kairala	2014-11-15	Feitas algumas alterações sintáticas no documento	12	91	187
53	Thiago Kairala	2014-11-15	Inserido especificações suplementares.	10	39	4
54	Thiago Kairala	2014-11-15	Inseridos Requisitos não funcionais.	5	43	12
55	Thiago Kairala	2014-11-15	Inserido requisitos funcionais	6	37	6
56	Thiago Kairala	2014-11-16	Alterada tabela de pontuação de acordo com os critérios do cliente	4	74	53

Sumário

Lista de Figuras

Lista de Tabelas

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. 20

- *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 possivel soluçao bem sucedida. A utilização do framework garante maior facilidade no entendimento do contexto do cliente.

- *Framework de Necessidades*

  Consiste em uma técnica para oganizar 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 partipantes 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 iriamos 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 Figura 1

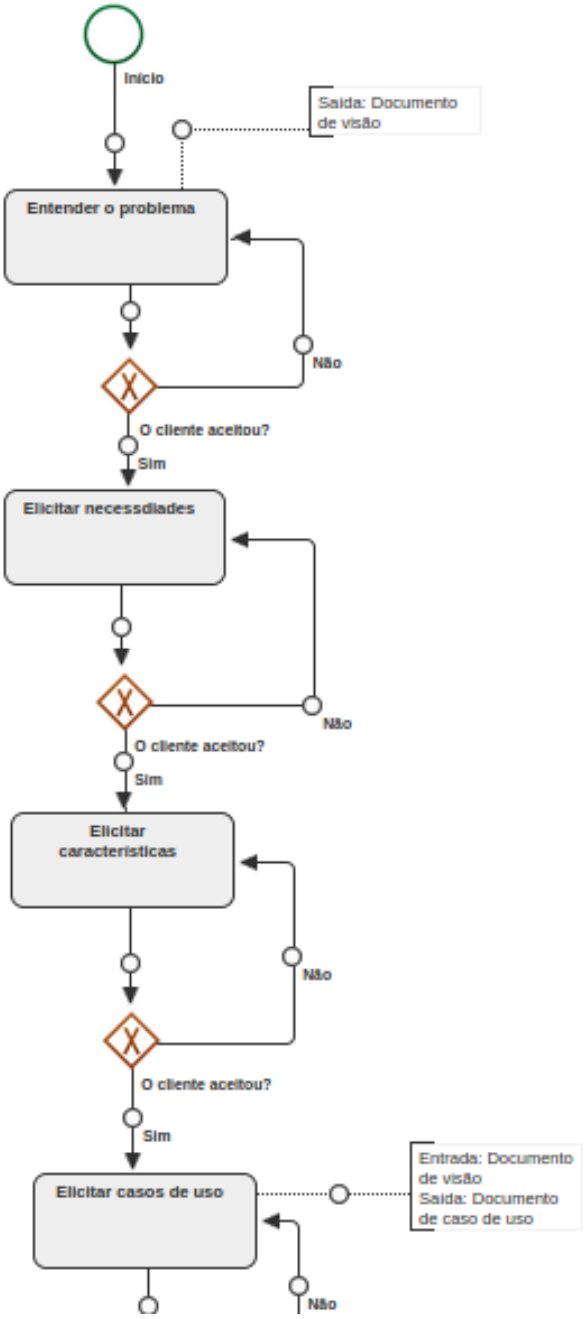

Figura 1. Modelagem Parte 1

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 Figura 2.

O detalhamento dos casos de uso está presente na Sessão 6 deste documento, assim como os roads maps se encontram na Sessão 4 .

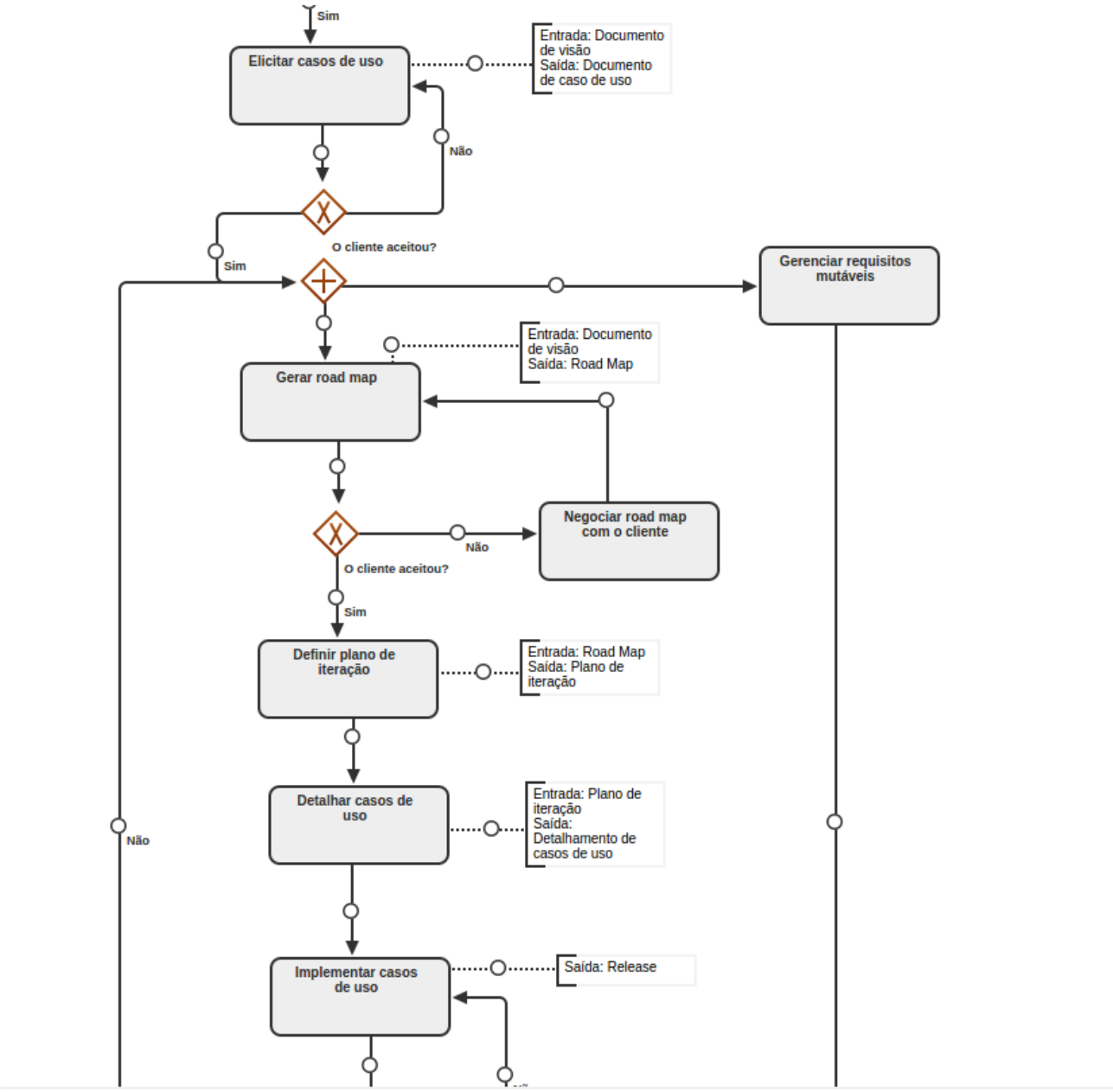

Figura 2. Modelagem Parte 2

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.

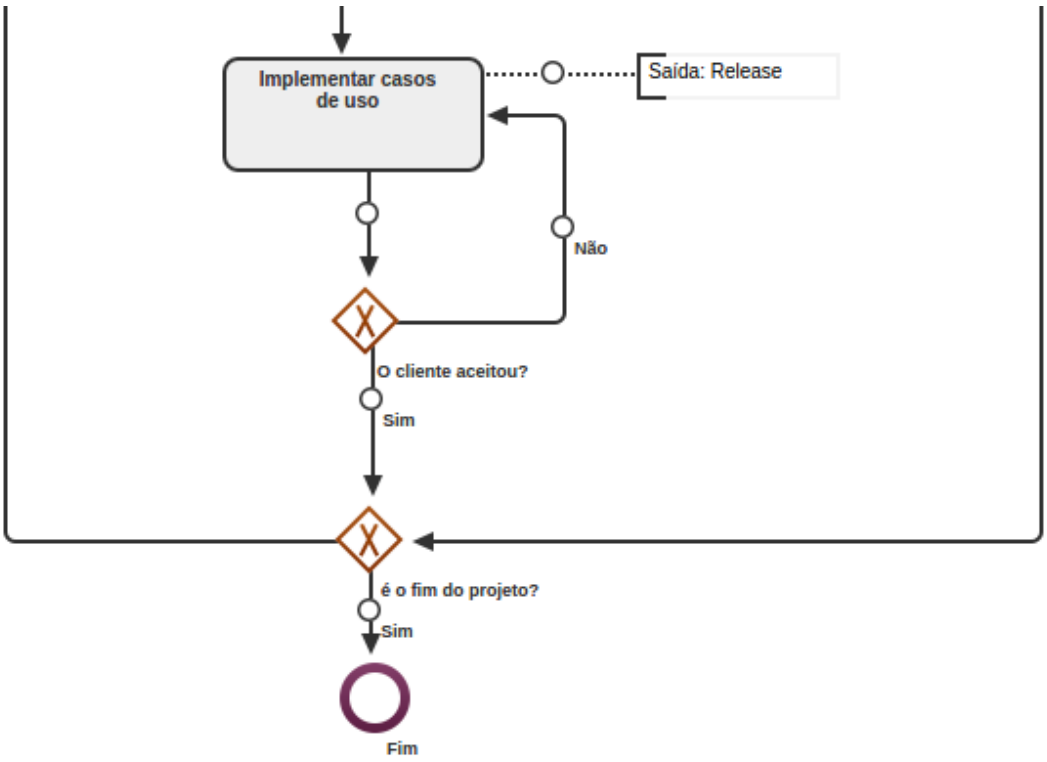

Figura 3. Modelagem Parte 3

3 Documento de visão

O documento de visão tem como objetivo definir uma visão geral do projeto, apresentar os problemas, os requisitos funcionais, não funcionanis, 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 díficil 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 serão organizados os requisitos do sistema, e a escolhida para o projeto esta ilustrada na Figura 4.

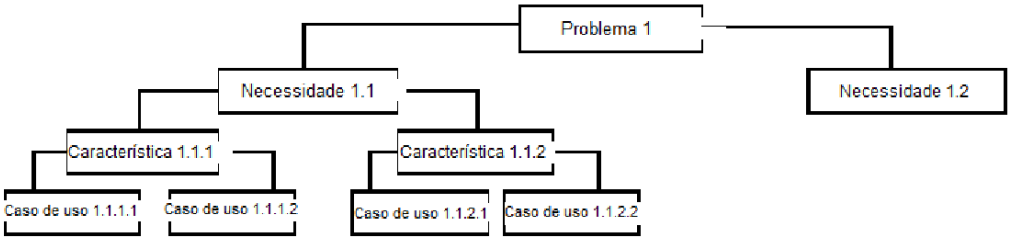

Figura 4. Matriz de rastreabilidade

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	Critérios 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 *software*.	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 *software*, 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.
Programador	Pessoa com capacidade em linguagens e lógica de programação	Implementar o sistema utilizando as técnologias definidas	Implementação do sistema de acordo com os requisitos levantados e cadastrados na ferramenta

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.

Para o mapeamento do problema principal e suas causas, foi utilizada a técnica do *Diagrama de Ishikawa*, que se encontra na Figura 5.

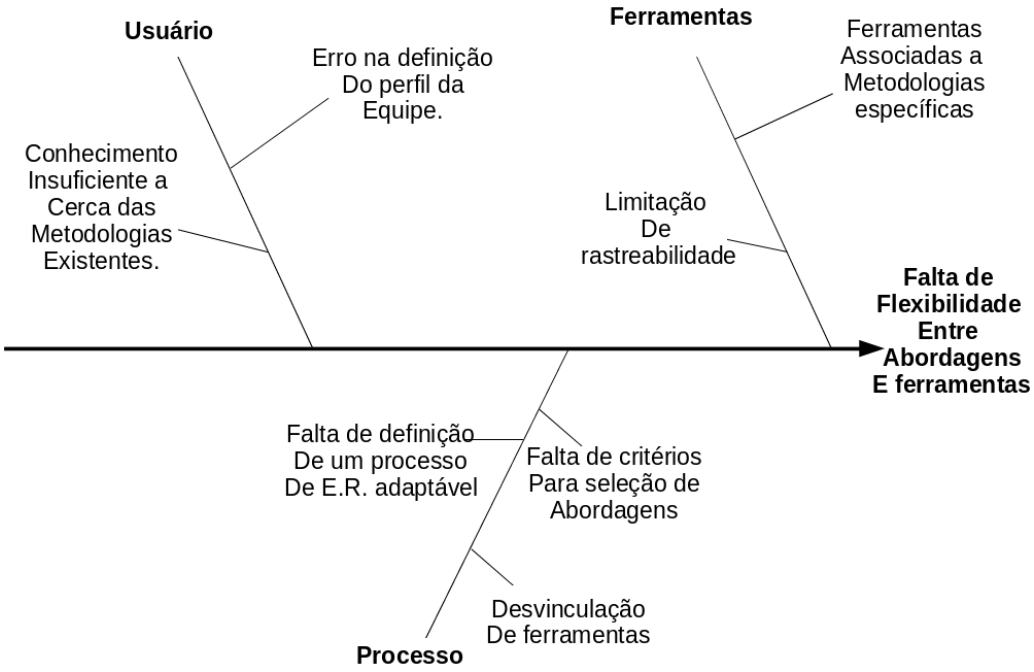

Figura 5. Diagrama de Ishikawa

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 e qual seria uma solução bem sucedida, o Framework está retratado na Tabela 3.

O Problema:	Falta de flexibilidade entre Abordagens e Ferramentas.
Afeta:	Todos os desenvolvedores de *software* 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 flexivel, 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
Utilização de ferramentas que se adequem as metodologias.	Ferramentas associadas a metodologias específicas.	Equipe utiliza mais de uma ferramenta para abranger as abordagens utilizadas.	Criação de uma ferramenta que seja flexível para qualquer metodologia, abrangendo todas as agordagens e até a mesclagem das mesmas.
Apoio a utilização de uma rastreabilidade organizada e eficiente em qualquer abordagem.	Limitação de ratreabilidade.	A equipe precisa criar sua rastreabilidade sem o apoio de uma ferramenta flexível.	Criação de uma ferramenta que gere a rastreabilidade dos requisitos de forma organizada e eficiente para qualquer abordagem.
Obter critérios fixos que redirecionem o projeto para a abordagem mais adequada.	Falta de critérios para seleção de abordagens.	Equipe precisa estudar as características do projeto e decidir qual a abordagem mais adequada.	Criação de uma ferramenta que recolha as características do projeto e apresente a abordagem mais adequada.
Obter um processo de *E.R.* adaptável a qualquer abordagem.	Falta de definição de um processo de *E.R.* adaptável.	Utilização de um processo inflexível e voltado apenas para uma abordagem.	Criação de uma ferramenta que gerencie processos flexíveis.
Gerar documentação de qualidade e fácil entendimento	Dificuldade em gerar documentação para pessoas de fora da equipe	Utilização de ferramentas a parte para gerar Diagramas de Caso de Uso e *Diagramas de Ishikawa*.	Integração da documentação na ferramenta de requisitos.

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 Requisitos Funcionais

Requisitos funcionais são as características do sistema associadas às funcionalidades do sistem em sí, como por exemplo o que o sistema deve fazer e como deve se comportar. Os requisitos funcionais estão listados abaixo.

- RF01 - O sistema deverá manter requisitos em qualquer metodologia selecionada;
- RF02 - O sistema deverá calcular a metodologia idela para o projeto;
- RF03 - O sistema deverá permitir ao usuário escolher a própria metodologia;
- RF04 - O sistema deverá pedir senha para o usuário acessar seus projetos;
- RF05 - O sistema deverá diferenciar usuários pelo nível de acesso em projetos;
- RF06 - O sistema deverá controlar quais usuários estão em cada projeto;
- RF07 - O sistema deverá ter um menu dropdown em todas as páginas para o usuário acessar seus projetos;
- RF08 - O sistema deverá ter um link em todas as páginas para criação de novos projetos;
- RF09 - O sistema deverá manter o controle das mudanças;
- RF10 - O sistema deverá gerar metodologias novas caso necessário.

3 .6 Recursos do Produto

Os recursos do produto são as funcionalidades do sistema, o que futuramente será transformado em casos de uso, estes recursos estão organizados de acordo com a rastreabilidade proposta na Figura 15 do relatório de projeto realizado anteriormente, e serão detalhados no documento de casos de uso, presente na Sessão 6 deste documento. Os recursos estão dividios na rastreabilidade ilustrada na Figura 6.

Problema	Necessidade	Características	Casos de Uso
P1	N1.1	C1.1.1	UC1.1.1.1
			UC1.1.1.2
			UC1.1.1.3
			UC1.1.1.4
		C1.1.2	UC1.1.2.1
			UC1.1.2.2
			UC1.1.2.3
			UC1.1.2.4
	N1.2	C1.2.1	UC1.2.1.1
			UC1.2.1.2
		C1.2.2	UC1.2.2.1
			UC1.2.2.2
	N1.3	C1.3.1	UC1.3.1.1
	N1.4	C1.4.1	UC1.4.1.1
			UC1.4.1.2
	N1.5	C1.5.1	UC1.5.1.1
			UC1.5.1.2
		C1.5.2	UC1.5.2.1
			UC1.5.2.2

Figura 6. Tabela de rastreabilidade

Seguem os recursos do sistema, apresentados utilizando a ratreabilidade apresentava na Figura 6:

3 .6.1 Problema 1 - Falta de flexibilidade entre abordagens e ferramentas

O problema 1, gera algumas necessidades, que serão colocadas a seguir.

3 .6.1.1 Necessidade N1.1 - Utilização de ferramentas que se adequem as metodologias

3 .6.1.1.1 Característica C1.1.1 - Manter metodologias tradicionais

- Caso de uso UC1.1.1.1 - Manter Problema;

  Este caso de uso será realizado pelo engenheiro de requisitos, e tem como objetivo cadastramento, remoção, atualização e leitura dos problemas na ferramenta.

- Caso de uso UC1.1.1.2 - Manter Necessidades;

  Este caso de uso será realizado pelo engenheiro de requisitos, e tem como objetivo o cadastramento, remoção, atualização e leitura das necessidades na ferramenta.

- Caso de uso UC1.1.1.3 - Manter Características;

  Este caso de uso será realizado pelo engenheiro de requisitos, e tem como objetivo o cadastramento, remoção, atualização e leitura das características na ferramenta.

- Caso de uso UC1.1.1.4 - Manter Casos de Uso.

  Este caso de uso será realizado pelo engenheiro de requisitos, e tem como objetivo o cadastramento, remoção, atualização e leitura dos casos de uso.

3 .6.1.1.2 Característica C1.1.2 - Manter metodologias ágeis

- Caso de uso UC1.1.2.1 - Manter Temas de Investimento;

  Este caso de uso será realizado pelo nível de portfólio do projeto, e tem como objetivo o cadastramento, remoção, atualização e leitura dos temas de investimento do projeto na ferramenta.

- Caso de uso UC1.1.2.2 - Manter Épicos;

  Este caso de uso será realizado pelo nível de portfólio do projeto, e tem como objetivo o cadastramento, remoção, atualização e leitura dos épicos do projeto na ferramenta.

- Caso de uso UC1.1.2.3 - Manter Features;

  Este caso de uso será realizado pelo nível de programa do projeto, e tem como objetivo o cadastramento, remoção, atualização e leitura das features do projeto na ferramenta.

- Caso de uso UC1.1.2.4 - Manter Histórias de Usuário.

  Este caso de uso será realizado pelo nível de time do projeto, e tem como objetivo o cadastramento, remoção, atualização e leitura dos temas de investimento do projeto na ferramenta.

3 .6.1.2 Necessidade N1.2 - Apoio a utilização de uma rastreabilidade organizada e eficiente em qualquer abordagem

3 .6.1.2.1 Característica C1.2.1 - Manter informações sobre requisitos

- Caso de uso UC1.2.1.1 - Manter Atributos;

  Este caso de uso será realizado ou pelo engenheiro de requisitos em uma metodologia tradicional, ou pelo nível de portfólio em metodologias ágeis, e tem como objetivo o cadastramento, remoção, atualização e leitura dos atributos no sistema, sendo atributos características dos requisitos.

- Caso de uso UC1.2.1.2 - Manter *Roadmaps*.

  Este caso de uso será reaelizado ou pelo engenheiro de requisitos em uma metodologia tradicional, ou pelo nível de portfólio em metodologias ágeis, e tem como objetivo o cadastramento, remoção, atualização e leitura dos *roadmaps* do projeto, sendo eles o uma priorização dos requisitos que devem ser implementados.

3 .6.1.2.2 Característica C1.2.2 - Manter relação entre Requisitos

- Caso de uso UC1.2.2.1 - Manter rastreabilidade horizontal entre os requisitos;

  Este caso de uso se refere ao ato do sistema manter as dependências dos requisitos com outros requisitos do mesmo nível, por exemplo um caso de uso que depende que um outro tenha sido feito já.

- Caso de uso UC1.2.2.2 - Manter rastreabilidade vertical entre os requisitos.

  Este caso de uso se refere ao ato do sistema manter as dependências dos requisitos com os requisitos de outros níveis, como por exemplo, quais necessidades estão ligadas a um problema.

3 .6.1.3 Necessidade N1.3 - Obter critérios fixos que direcionem o projeto para abordagem mais adequada

3 .6.1.3.1 Característica C1.3.1 - Auxiliar na escolha da metodologia

- Caso de Uso UC1.3.1.1 - Definir Metodologia.

  Este caso de uso se refere ao ato do sistema definir a metodologia a ser adotada no projeto, de acordo com os dados entrados pelo engenheiro de requisitos.

3 .6.1.4 Necessidade N1.4 - Obter um processo de Engenharia de Requisitos adaptável a qualquer abordagem

3 .6.1.4.1 Característica C1.4.1 - Criar processos Híbridos

- Caso de uso UC1.4.1.1 - Definir "hibridez" do projeto;

  Este caso de uso se refere em montar uma metodologia própria para o projeto a ser desenvolvido, mantendo quais características de cada metodologia, ágil e tradicional, é mais adequada para cada característica do projeto.

- Caso de uso UC1.4.1.2 - Manter processos híbridos.

  Este caso de uso se refere ao ato do engenheiro de requisitos realizar o cadastramento, remoção, atualização e leitura dos requisitos de projetos híbridos.

3 .6.1.5 Necessidade N1.5 - Gerar documentação de qualidade e fácil entendimento

3 .6.1.5.1 Característica C1.5.1 - Gerar e manter diagramas

- Caso de uso UC1.5.1.1 - Gerar *Diagrama de Ishikawa*;

  Este caso de uso se refere a geração do *Diagrama de Ishikawa* pelo engenheiro de requisitos para facilitar na identificação do problema a ser solucionado.

- Caso de uso UC1.5.1.2 - Gerar Diagramas de Casos de Uso.

  Este caso de uso se refere a geração do Diagrama de caso de uso por parte do engenheiro de requisitos em metodologias tradicionais.

3 .6.1.5.2 Característica C1.5.2 - Controlar projeto por toda sua duração

1. Caso de uso UC1.5.2.1 - Gerar plano de iteração;

   Este caso de uso se refere a geração dos planos de iteração, tanto em metodologias ágeis, *backlog*, quanto em metodologias tradicionais, o plano de iteração.

2. Caso de uso UC1.5.2.2 - Controlar histórico de versão.

   Este caso de uso se refere ao controle das alterações realizadas nos projetos, e a possibilidade de voltar atrás em qualquer alteração, além do histórico de quem realizou cada alteração.

3 .7 Restrições

- Técnica A ferramenta poderá ser executada pelos navegadores Google Chrome versão 37.0.2062.120 ou superior Firefox versão 33.0 ou superior, não sendo possível sua utilização no Internet Explorer ou Safari.

- Tempo A primeira release da ferramenta deverá ser entregue até no máximo no dia 22 de novembro de 2014.

3 .8 Requisitos não funcionais

Requisitos não funcionais são características que não são funcionalidades do sistema em si, estão relacionados com aspectos como segurança, usabilidade, confiabilidade e performace [Araujo et al.].

A seguir estão listados os requisitos não funcionais do sistema em desenvolvimento.

- A ferramenta necessitará de conexão com a internet;

- Deverá manter a segurança dos dados do sistema;

- Deverá ser possível acessá-la tanto de computadores como aparelhos móveis;
- A ferramenta deverá ser *open source* e sobre a licença *GNU Affero General Public License* descrita em http://www.gnu.org/licenses/agpl-3.0.html;
- Durante o projeto apenas deverão ser usados ferramentas *open source*.

3 .9 Atributos do Recurso

Atributos de recursos são basicamente descrições dos requisitos em alguma área em específico. Durante o projeto foram utilizados os atributos de arquitetura, prioridade e status, a tabela contendo estes atributos está ilustrada no relatório de projeto feito anteriormente.

4 RoadMap

Roadmaps são uma priorização dos recursos do sistema, para definir por qual requisito a implementação terá início.

Para gerar o *roadmap* foi gerada uma pontuação nos atributos dos recursos presentes nas Tabelas **??** e **??**, mostrada na Tabela 5.

Atributo	**Classificação**	**Pontuação**
Prioridade	Alta prioridade	5
	Média prioridade	3
	Baixa prioridade	1
Arquitetura	Grande	7
	Média	5
	Baixa	3
	Nenhuma	1

Tabela 5. Pontuação dos Atributos

Utilizando a Tabela 5, fomos capazes de fazer uma relação numérica para pontuar cada um dos recursos, apresentados na Sessão 3 .6 deste documento, e fazer a escolha de qual deve ser implementado primeiro, esta relação está apresentada na Tabela 6.

Recurso	**Atributo de prioridade**	**Atributo de arquitetura**	**Pontuação final**
Definir Metodologia	Alta Prioridade	Risco Alto	12
Manter Problema	Baixa Prioridade	Risco Baixo	4
Manter Necessidades	Baixa Prioridade	Risco Baixo	4
Manter Características	Baixa Prioridade	Risco Baixo	4
Manter Casos de Uso	Baixa Prioridade	Risco Baixo	4
Manter Temas de Investimento	Baixa Prioridade	Risco Baixo	4
Manter Épicos	Baixa Prioridade	Risco Baixo	4
Manter Features	Baixa Prioridade	Risco Baixo	4
Manter Histórias de Usuário	Baixa Prioridade	Risco Baixo	4
Manter Atributos	Baixa Prioridade	Nenhum Risco	2
Manter Rastreabilidade Horizontal entre os Requisitos	Média Prioridade	Risco Baixo	6
Manter Rastreabilidade vertical entre os Requisitos	Média Prioridade	Risco Baixo	6
Gerar *Diagrama de Ishikawa*	Baixa Prioridade	Risco Médio	6
Manter Atores do Projeto	Baixa Prioridade	Risco Baixo	4
Gerar Diagramas de Casos de Uso	Média Prioridade	Risco Médio	8
Manter *Roadmaps*	Média Prioridade	Risco Médio	8
Gerar plano de iteração	Baixa Prioridade	Nenhum Risco	2
Definir “hibridez” do projeto	Alta Prioridade	Risco Alto	12
Manter processos híbridos	Alta Prioridade	Risco Médio	10
Controlar histórico de versão	Baixa Prioridade	Risco Médio	7

Tabela 6. Pontuação dos recursos

Utilizando os dados apresentados na Tabela 6, podemos então gerar um *rank* da ordem em que as funcionalidades devem ser implementadas, e a ordem deve ser de acordo com a lista a baixo:

1. **Primeira prioridade:**
    - Manter problema;
    - Manter Necessidade;
    - Manter Características;
    - Manter Casos de uso;
    - Manter Temas de investimento;
    - Manter Épicos;
    - Manter Features;
    - Manter Histórias de usuário;
    - definir “hibridez” do projeto.
2. **Segunda prioridade:**
    - Definir metodologia;
    - Manter rastreabilidade de requisitos;
3. **Terceira prioridade:**
    - Gerar *Diagrama de Ishikawa*;

- Gerar Diagrama de Caso de Uso;
- Manter *roadmaps*.

4. **Quarta prioridade:**
   - Manter Atributos.
5. **Quinta prioridade:**
   - Manter atores do projeto;
   - gerar planos de iteração.
6. **Sexta prioridade:**
   - Controlar histórico de versão.

Após estudo e análise das metodologias possíveis para desenvilvimento da ferramenta, escolheu-se a utilização da metodologia Ágil, graças ao pequeno tempo para desenvolvimento, a disponibilidade do cliente e o tamanho da equipe. Como o tempo de desenvolvimento será bastante curto, não serão implementados todos os requisitos do sistema. Dessa forma, apresenta-se o *roadmap* utilizado nas iterações que serão realizados na Tabela 7.

	Iteração 1	**Iteração 2**
Casos de uso	• Definir metodologia; • Manter Temas de investimento; • Manter Épicos.	• Manter Features; • Manter Histórias de usuário.

Tabela 7. *Roadmap*

Este *roadmap* será utilizado mais a frente, na Sessão de 6 para auxiliar quais casos de uso devem ou não ser detalhados durante o processo de desenvolvimento.

5 Especificações suplementares

As especificações suplementares listam todas as definições dos sistema que não estão incluídas no modelo de Caso de uso, ou seja, não estão relacionadas com a funcionalidade em sí do sistema, **??**.

5 .1 Características do sistema

As características do sitema são representadas nesta sessão são as definições do que mais adiante será transformado em requisitos não funcionais.

5 .1.1 Usabilidade

A ferramenta R.A.D.I.T. deve ser de fácil utilização, a ponto de não ser necessário treinamento do usuário para a utilização das funcionalidades básicas, e deve ser intuitiva, para que após um mês, no máximo, o usuário alcance uma produtividade boa em manipular os requisitos e utilizar o sistema.

5 .1.2 Confiabilidade

A ferramenta R.A.D.I.T. deve estar disponível pelo menos 95% do tempo, desonsiderando erros do servidor no qual será instalada pelo usuário, e deve ser possuir resiliência suficiente para que não seja necessário reinicialização do sistema a cada quebra.

Infelizmente, não é possível gerar um sistema 100% *anti-quebra*, e sabemos que a ferramenta irá passar por falhas. Porém estabelecemos que o tempo mínimo entre falhas do sistema deverá ser de 1 mês.

O sistema também deverá ser confiável em relação à invasões, devendo ser capaz de impedir os ataques básicos ao sistema, e ao seu banco de dados.

5 .1.3 Desempenho

A ferramenta será considerada com um desempenho bom quando qualquer uma de suas páginas não levar mais de dois segundos para ser carregada, e em média levar apenas um segundo.

Além do tempo de resposta das páginas outro quesito para a ferramenta possuir um bom desempenho é o número de consultas no banco de dados por funcionalidade, não devendo ser maior que uma.

O sistema deverá ser capaz também de receber um total de mil usuários simultâneos ter influência na regra do tempo de resposta de cada uma das páginas.

5 .2 Requisitos não funcionais

Os requisitos não funcionais partem das características do sistema, citados na Sessão 5 .1, e são listados abaixo.

1. Usabilidade
    - RNF01 - O usuário não deve necessitar de treinamento para utilizar a ferramenta;
    - RNF02 - O usuário não deve levar mais de um mês para estar totalmente produtivo na utilização da ferramenta.
2. Confiabilidade
    - RNF03 - O sistema deve estar disponível 95% do tempo;
    - RNF04 - O sistema deve ser resiliente;
    - RNF05 - O sistema deve ter um espaço mínimo de 1 mês entre uma falha e outra;
    - RNF06 - O sistema deve ser capaz de resistir à invasões conhecidas.
3. Desempenho
    - RNF07 - O sistema deve ter um tempo de resposta em qualquer página de no máximo dois segundos e em média um segundo;
    - RNF08 - O sistema deve ter no máximo 1 consulta ao banco de dados por funcionalidade;
    - RNF09 - O sistema deve suportar até mil usuários sem redução no tempo de resposta das páginas.

6 Documento de casos de uso

O documento de casos de uso tem como finalidade o detalhamento a fundo dos recursos do programa, aqui chamados de caso de uso, listados na Sessão 3 .6 deste documento, colocando todas as suas características, restrições e caminhos possíveis.

6 .1 Identificação dos atores

Neste contexto, atores são representações genéricas de usuários do sistema, podende ser qualquer utilizador, sem se preocupar com nome do executor, apenas com sua função, esses atores estão listados na Tabela 8.

Atores	**Descrição**
Engenheiro de Requisitos	Responsável, em metodologias tradicionais, pela elicitação dos requisitos e pela manutenção da rastreabilidade do sistema
Analista de Requisitos	Responsável, em metodologias tradicionais, gerência dos requisitos
Product Owner	Responsável, em metodologias ágeis, por escrever histórias de usuário
Equipe de portfólio	Responsável, em metodologias ágeis, por gerir a parte do portfólio, como temas de investimento e épicos
Equipe de programa	Responsável, em metodologias ágeis, por gerir as features do sistema e manter a entrega das releases em dia
Time	Responsável, em metodologias ágeis, por implementar as histórias de usuário
Cliente	Responsável por validar os requisitos

Tabela 8. Atores do sistema

6 .2 Diagrama de casos de uso

6 .3 Detalhamento dos casos de uso

Detalhamentos de casos de uso seve para definir o que cada caso de uso fará, quem irá realizá-lo, e como ele irá responder a falhas caso haja, e todos os seus caminhos possíveis.

A seguir estão os detalhamentos dos casos de uso que serão implementados nas sprints 1 e 2, detalhadas na Tabela 7.

6 .3.1 Caso de Uso - UC1.3.1.1 - Definir Metodologia

6 .3.1.1 Descrição Este caso de uso especifica a ação do sistema de, dada as informações solicitadas, selecionar a melhor rota possível para o desenvolvimento do projeto, podendo o usuário, ao final do questionário, decidir se irá seguir ou não a rota sugerida, e então preparar a ferramenta para a metodologia escolhida.

1. Atores Engenheiro de requisitos.
2. Pré-condições Não existe pré condições para este caso de uso.
3. Pós-condições A rota a ser utilizada deve estar definida ao final da execução deste caso de uso.
4. Requisitos Funcionais
    - RF02 - O sistema deverá calcular a metodologia idela para o projeto;
    - RF03 - O sistema deverá permitir ao usuário escolher a própria metodologia;
    - RF07 - O sistema deverá ter um menu dropdown em todas as páginas para o usuário acessar seus projetos;
    - RF08 - O sistema deverá ter um link em todas as páginas para criação de novos projetos;

- RF10 - O sistema deverá gerar metodologias novas caso necessário.

5. Requisitos Não Funcionais

   - RNF01 - O usuário não deve necessitar de treinamento para utilizar a ferramenta;
   - RNF02 - O usuário não deve levar mais de um mês para estar totalmente produtivo na utilização da ferramenta.
   - RNF07 - O sistema deve ter um tempo de resposta em qualquer página de no máximo dois segundos e em média um segundo;
   - RNF08 - O sistema deve ter no máximo 1 consulta ao banco de dados por funcionalidade;

6 .3.1.2 Fluxo básico

1. Ator decide criar um novo projeto;
2. Sistema apresenta um questionário para recolher informações doprojeto; As perguntas são:
   - Qual o tamanho da equipe;
   - O projeto terá troca de quipe durante seu desenvolvimento;
   - O cliente solicita documentação extensa.
3. Ator responde questionário;
4. Sistema calcula estisticamente qual rota deve ser utilizada, de acorodo com as respostas do ator;
5. Sistema apresenta ao ator a escolha da metodologia;
6. Usuário aceita a metodologia;
7. Sistema prepara a ferramenta para utilização da metodologia escolhida.

6 .3.1.3 Fluxo alternativo A

1. No passo 4 do fluxo básico, caso haja um empate entre metodologias;
2. Sistema apresenta aos ao ator as metodologias empatadas e suas características;
3. Ator escolhe a metodologia que deseja;
4. O fluxo retorna para o passo 7 do fluxo básico.

6 .3.1.4 Fluxo alternativo B

- No passo 6 do fluxo básico, caso o ator não aceite a metodologia proposta pelo sistema;
- Ator rejeita a opção da metodologia escolhida pelo sistema;
- Sistema apresenta todas as opções de metodologias cadastradas para que o usuário possa escolher;
- retorna para o passo 7 do fluxo básico.

6 .3.2 Caso de Uso - UC1.4.1.1 - Definir “hibridez” do projeto

6 .3.2.1 Descrição Este caso de uso auxilia a descobrir quão hibrido é o projeto proposto pelo cliente quais as caracteristicas ágeis e quais caracteristícas tradicionais que irão compor a "hibridez"do projeto.

1. Atores Engenheiro de requisitos.
2. Pré-condições Não existe pré condições para este caso de uso.
3. Pós-condições ao final do caso de uso deverão ser mostrados quais caracteristicas das abordagens o projeto irá herdar a fim de ver quão hibrido é o projeto"

Referências Bibliográficas

[Araujo et al.]ARAUJO, A. C. M. de et al. Requisitos não funcionais.

[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/inc

[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, porêm, 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 nao te agradram?

    R: Não me agradam?

    A necessidade de criação ou não de uma ferramenta não vem da necessidade de ferramente 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 ferramente, poderá ser desenvolvido um plug-in para alguma existente que seja possível tal adaptação, ou o desenvolvimento de uma nova ferramente 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.

- **QUESTIONARIO**
  - **Descrição do questionario**

    Com a finalidade de auxiliar na escolha da abordagem do projeto do foram analizados artigos e tendo como base os mesmos foram definidas algumas caracateristicas de cada metodologia ágil e tradicional afim de elaborar um questionario que auxilie nessa tomada de decisão.

    Obs 1- algumas questões terão peso maior que as outras por serem caracteristicas mais influentes em cada métodologia.

    Obs 2- as questões serão pontuadas como:

    pouco importante - 1

    importante - 2

    Muito importante -3

    ao final do questionario a abordagem que tiver o maior Score incluindo o peso de cada pergunta será sugerido ao usuario uma abordagem em que o mesmo pode decidir utiliza-la ou nao