



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

Relatório do Trabalho 1

Autor: Daniel Lucas Assunção Teles
Felipe Duerno
João Vítor de Brito Fernandes
Luan Guimarães
Orientador: Profa. Elaine Venson

Brasília, DF
Maio de 2016



Daniel Lucas Assunção Teles
Felipe Duerno
João Vítor de Brito Fernandes
Luan Guimarães

Relatório do Trabalho 1

Relatório de Projeto submetido à disciplina de Requisitos de Software do curso de Engenharia de Software na Faculdade UnB Gama da Universidade de Brasília como requisito parcial para composição da nota final.

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA

Orientador: Profa. Elaine Venson

Brasília, DF
Maio de 2016

Lista de ilustrações

Lista de tabelas

Sumário

1	INTRODUÇÃO	7
1.1	Contexto do cliente	7
2	ABORDAGEM DA ENGENHARIA DE REQUISITOS	9
2.1	SAFe	9
2.2	RUP	10
2.3	Modelo de maturidade	10
2.3.1	Modelo CMMI(Capability Maturity Model Integration)	11
2.3.2	Modelo MPS-BR (Melhoria de Processo do Software Brasileiro)	11
2.4	Contexto dentro do projeto	13
2.5	Justificativa	13
3	PROCESSO DA ENGENHARIA DE REQUISITOS	15
3.1	Big picture do projeto	15
3.2	Papéis no processo	15
3.3	Atividades e artefatos do processo	15
4	ELICITAÇÃO DE REQUISITOS	17
4.1	Técnicas para a elicitação de requisitos	17
4.1.1	Brainstorming	17
4.1.2	Entrevista	17
4.1.3	Workshop	18
5	GERENCIAMENTO DE REQUISITOS	19
5.1	Atributos de um requisito	19
5.1.1	Prioridade	19
5.1.2	Complexidade	19
5.1.3	Responsável	19
5.1.4	Status	19
5.2	Rastreabilidade de requisitos	19
5.3	Gestão de requisitos	19

1 Introdução

1.1 Contexto do cliente

A empresa júnior de engenharia eletrônica da Universidade de Brasília - Faculdade do Gama, a Eletrojun, foi a primeira empresa júnior da FGA. Por ser a empresa mais antiga ela é a que tem maior demanda de serviços e o maior número de clientes. Isso gerou uma carência no que diz respeito ao gerenciamento de clientes e seus respectivos projetos. Desde o primeiro contato com o cliente até a entrega do produto final, o projeto passa por várias etapas em diferentes setores da empresa. Isso requer um processo minucioso, que se atente com cada detalhe, cada iteração.

A Eletrojun possui uma organização estrutural linear e vertical. No topo da empresa está o presidente, logo abaixo vem os diretores, sendo eles: administrativos, financeiros, de gestão, marketing e projetos. Abaixo dos diretores estão os gerentes, dispostos em 3 áreas principais, que são: administrativa, financeira e de auditoria. Abaixo da gerência estão todos os outros membros da instituição. Este tipo de estrutura torna bem claro para cada setor quais atividades devem ser desenvolvidas.

2 Abordagem da Engenharia de Requisitos

“Um conjunto de atividades utilizadas para identificar e comunicar a finalidade de um sistema desoftware, e o contexto no qual sera usado. Assim, a ER atua como a ponte entre as necessidades reais dos usuarios, clientes, e outros grupos afetados por um sistema de software , e as potencialidades e oportunidades oferecidas pela tecnologia”. (EASTERBROOK, 2004)

“Nos dias atuais, a necessidade de automatizacao de processos se faz cada vez mais presente. Processos sao essenciais dentro de qualquer organizacao em qualquer area de atuacao, inclusive em uma das areas mais recentes, que a producao de software. A definicao de processos de desenvolvimento de software vem com o objetivo de aumentar a produtividade e diminuir os riscos de um projeto”. (VIEIRA, 2003)

“Com o grande crescimento da demanda na producao de software na atualidade os prazos estao cada vez mais curtos e cobranca relacionada a qualidade estao cada vez maiores. A qualidade de um produto de software esta diretamente ligada a melhor forma de atender as necessidades do cliente, o que torna um produto totalmente dinamico. Uma vez que a natureza do produto e dinamica existem grandes dificuldades no gerenciamento do desenvolvimento, por exemplo: a essencia volatil dos requisitos do cliente”. (TAVARES, 2008).

Dados estes fatores, foram criados vários processos dentro da engenharia de software, a fim de amenizar esses problemas, diminuir o tempo gasto e os custos de desenvolvimento. Por exemplo, os processos: cascata, interativo, espiral e incremental.

Cada processo se encaixa melhor em um determinado contexto, geralmente o processo da ER é adaptado de acordo com a equipe, com o cliente e com o projeto. Dado o contexto do trabalho a ser confeccionado nessa disciplina foram propostas duas abordagens, sendo elas: Scaled Agile Framework(SAFe) e o Rational Unified Process(RUP). A seguir daremos uma breve explicação sobre como funciona cada abordagem e será exposta a justificativa da escolha da abordagem e suas atividades.

2.1 SAFe

O Scaled Agile Framework (SAFe) é um processo que provê uma receita para a aplicação de práticas e princípios ágeis e enxutos em uma escala empresarial. Assim como o Scrum está para uma equipe ágil o SAFe está para um empreendimento ágil. Possui uma abordagem focada na gestão e captura dos requisitos, em vez de no design e implementação.

É composto essencialmente por três níveis: o nível de Time, o nível de Programa e o nível de Portfólio. Para cada nível, existe um nível diferente de narrativa de requisitos: épicos para o Portfólio, features para o Programa e histórias de usuários para o Time. A partir da versão 4.0 deste framework, foi adicionado um novo nível, facultativo, chamado Fluxo de Valor.

O SAFe possui nove princípios fundamentais, baseados nos princípios ágeis, que orientam os papéis e práticas que o tornam efetivo. Esses princípios são:

- Tenha uma visão econômica
- Tenha um pensamento sistêmico
- Assuma variabilidade e preserve opções
- Construa incrementalmente com ciclos de aprendizagem integrados e rápidos
- Baseie os marcos em avaliação objetiva dos sistemas
- Descentralize a tomada de decisão
- Desbloqueie a motivação intrínseca aos profissionais do conhecimento
- Tenha cadência; sincronize com planejamento cross-domain
- Visualize e limite o trabalho em progresso, reduza a carga e gerencie o comprimento da fila

2.2 RUP

2.3 Modelo de maturidade

Os Modelos de Maturidade de Processos são um referencial usado para:

- Avaliar a capacidade de processos na realização de seus objetivos;
- Localizar oportunidades de melhoria de produtividade e qualidade e de redução de custos;
- Planejar e monitorar as ações de melhoria contínua dos processos empresariais.

2.3.1 Modelo CMMI(Capability Maturity Model Integration)

O CMMI é um modelo de referência que define práticas sejam elas genéricas ou específicas necessárias para o desenvolvimento e avaliação de maturidade no desenvolvimento de softwares em uma organização. As práticas que são abordadas neste modelo são: gerenciamento de requisitos, manipulação de riscos, medição de desempenho, planejamento de trabalho, tomada de decisão, entre outros. O modelo CMMI não pode ser considerado uma metodologia, pois não orienta como deve ser feito, e sim o que deve ser feito. Esse modelo foi desenvolvido pelo SEI(Software Engineering Institute) da Universidade Carnegie Mellon. É uma evolução do CMM, que foi baseado em algumas das ideias mais importantes dos movimentos de qualidade industrial das últimas décadas. No CMMI uma organização opta por duas representações para a melhoria dos seus processos: Por estágios e contínua.

2.3.2 Modelo MPS-BR (Melhoria de Processo do Software Brasileiro)

MPS-BR significa Melhoria de Processo do Software Brasileiro, criado pelo Softex e patrocinado pelo MCT. O modelo de maturidade de processos e desenvolvimento de software conhecido como CMMI-DEV foi adaptado para empresas brasileiras, em especial para micro, pequenas e médias empresas, dando origem ao MPS-BR. Essa adaptação foi necessária por que o CMMI-DEV prevê o amadurecimento dos processos em apenas cinco níveis. E com o passar do tempo percebeu-se a necessidade de uma funcionalidade mais gradual aqui no Brasil, por isso foi quebrado os cinco níveis do CMMI-DEV em sete, com vemos na figura 2:

Como ilustrado na imagem, o MPS-Br possui uma divisão semelhante ao CMMI, entretanto o mesmo encontra-se dividido nos níveis constituintes do A ao G, sendo o nível A o mais alto qualitativamente e o G o nível inicial(MPS de Software, São Paulo: SOFTEX, 2012).

- A. Em otimização
- B. Gerenciado quantitativamente
- C. Definido
- D. Largamente definido
- E. Parcialmente definido
- F. Gerenciado
- G. Parcialmente gerenciado

Os processos referentes a engenharia de requisitos estão descritos no níveis, G e D, correspondentes aos processos de "Gerência de requisitos respectivamente" e "Desenvolvimento de requisitos".

Gerência de requisitos

Como o nome já diz, tem como objetivo gerenciar os requisitos do produto e dos componentes do produto/projeto, além de identificar inconsistências entre requisitos, planos e produtos de trabalho do projeto.

Resultados esperados:

- **GRE 1** - O entendimento dos requisitos é obtido junto aos fornecedores de requisitos.
- **GRE 2** - Os requisitos são avaliados com base em critérios e objetivos e um comprometimento da equipe técnica com estes requisitos é obtido.
- **GRE 3** - A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida.
- **GRE 4** - Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos.
- **GRE 5** - Mudanças nos requisitos são gerenciadas ao longo do projeto.

Desenvolvimento de requisitos: Visa 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.
- **DRE 2** - Um conjunto definido de requisitos do cliente é especificado e priorizado a partir das necessidades, expectativas e restrições identificadas.
- **DRE 3** - Um conjunto de requisitos funcionais e não-funcionais, do produto que descrevem a solução do problema a ser resolvido, é definido e mantido a partir dos requisitos do cliente.
- **DRE 4** - Os requisitos funcionais e não-funcionais de cada componente do produto são refinados, elaborados e alocados.
- **DRE 5** - Interfaces internas e externas do produto e de cada componente do produto são definidas.
- **DRE 6** - Conceitos operacionais e cenários são desenvolvidos.

2.4 Contexto dentro do projeto

Neste projeto foi definida a utilização do modelo MPS-Br, levando em conta que ele é bem semelhante ao modelo CMMI e vai possibilitar a produção de resultados parecidos, foram observados outros critérios para a definição do modelo utilizado.

O MPS-Br foi a primeira opção apontada pelo grupo, o fato de ser um modelo brasileiro, voltado para a nossa realidade no que tange o desenvolvimento de software pesou bastante na hora da escolha. O MPS-Br, como dito na sessão 2.3.2, foi pensado para ser utilizado no contexto de micro, pequenas e medias empresas, esse foi outro fator importante na hora da escolha, uma vez que o nosso cliente é uma empresa júnior de engenharia eletrônica. Outro fator que vale a pena destacar é o fato de que o MPS-Br possui uma maior divisão de níveis de maturidade, o que torna o processo para atingir níveis maiores de maturidade algo mais simples, uma vez que por ter mais níveis o número de processos em cada nível é menor do que no CMMI.

2.5 Justificativa

A metodologia escolhida para a confecção deste trabalho foi uma abordagem ágil, fundamentada no SAFe. Esta escolha foi baseada nas características da equipe e do projeto em si, outro fator importante para a escolha desta metodologia foi o prévio relacionamento que um dos membros do grupo tinha com o nosso cliente, uma vez que o mesmo já havia trabalhado na empresa, além de conhecer a sua estrutura, seus processos e parte da equipe. Isso tornou o processo mais fluido, uma vez que não tivemos a necessidade de fazer uma pesquisa mais profunda a fim de conhecer melhor a empresa.(FINALIZAR/MELHORAR ESSA PARTE, MELHORAR BASTANTE)

3 Processo da Engenharia de Requisitos

O processo descrito a seguir foi modelado na ferramenta "Bizagi BPMN Modeler". Como dito anteriormente a abordagem a ser utilizada será uma abordagem ágil, fundamentada no SAFe(Scaled Agile Framework).

A estrutura dos tres niveis do SAFe foi mantida porque, segundo Leffingwell(2011), ao ir abaixando o nivel de abstracao dos requisitos mais gradativamente, e reduzido o nivel de especificacao precoce, reduzindo a sobrecarga ao gerenciar os requisitos. Isso tambem aumenta a agilidade do time por permitir a interpretacao dos requisitos de maneira mais facil para a implementacao (Leffingwell, 2011).

3.1 Big picture do projeto

3.2 Papéis no processo

Especialista do Negocio - É o stakeholder que detem o conhecimento do negocio, do contexto organizacional e da visao do produto.

Product Owner - É o membro do time que fica responsável pela definição das histórias e pela priorização do Team Backlog, além de participar do planejamento e validação da sprint definindo os seus objetivos.

Product Manager - De acordo com Leffingwell(2011), cabe ao PM: manter a visão e o program backlog, priorizar features, manter o roadmap, gerenciar o conteúdo da release e manter e priorizar o porfolio backlog. As atividades realizadas pelo PM acontecem nos níveis Portfolio e Program.

Scrum Master - O papel do Scrum Master e dar assistência para o resto da equipe a fim de extrair a máxima perfomance, ele é de certa forma o líder do time(Leffingwell, 2011).

Time - O time é composto por toda a equipe, desenvolvedores, designers e etc.

3.3 Atividades e artefatos do processo

Temas de investimento - Representam um conjunto de iniciativas guias para o investimento da instituição, seja em sistemas, produtos, aplicações ou serviços(Leffingwell, 2011).

Épicos - São iniciativas de desenvolvimento em larga escala que agregam valor a um tema de investimento(Leffingwell, 2011). São os de requisitos que possuem o mais alto nível no processo(Leffingwell, 2011).

Portfolio backlog - É o local onde os épicos são registrados, como um repositório de épicos.

Features - Atuam como pontes entre as necessidades dos stakeholders e os requisitos específicos no domínio da solução(Leffingwell, 2011).

Program backlog - É o local onde as features são registradas, como um repositório de features.

Roadmap - Consiste em uma série de releases com datas planejadas, cada uma pertinente a um tema, um conjunto de objetivos e uma feature priorizada. O roadmap nos dá uma ideia de como a instituição planeja mostrar valor com o decorrer do tempo(Leffingwell, 2011).

Visão - É um mecanismo utilizado para definir e comunicar a visão do sistema(Leffingwell, 2011). A visão de um sistema é composta por um conjunto de features que irão descrever as possibilidades do sistema, isto é, tudo aquilo que ele poderá oferecer ao usuário a fim de atender as necessidades dos envolvidos(Leffingwell, 2011).

Team backlog - Representa uma coleção de tudo que o time necessita para progredir aquela porção do sistema. Pode conter histórias de usuário ou enabler histories. Sendo que a maioria tem origem no program backlog, enquanto algumas são pertinentes a um contexto específico do time. O team backlog pertence ao PO, vale salientar que o fato de "pertencer"ao PO não significa que ele é o único que pode alterá-lo, mas é preferível que seja ele a fazer isso.(SAFe, 2015)

Sprint backlog - É o local onde serão armazenadas as histórias de usuário a serem realizadas naquela sprint.

Build - É o incremento de software que é gerado em cada sprint.

4 Elicitação de requisitos

A elicitação de requisitos é uma das mais importantes, senão a mais importante atividade quando se fala em desenvolvimento de software. Essa atividade consiste em adquirir o máximo possível de informações e conceitos em relação ao projeto, necessidades dos stakeholders, contexto, restrições, e problema a ser resolvido. Além de ser uma etapa importante para tornar o conhecimento entre o cliente e os desenvolvedores mais homogêneo, de forma que todos tenham o mesmo entendimento do problema a ser resolvido e de que forma ele deverá ser resolvido.

4.1 Técnicas para a elicitação de requisitos

Muitas vezes o cliente não consegue transmitir de forma clara o que deseja, a inexperiência ou falta de preparo do engenheiro de requisitos também é um fator que pode influenciar na hora de elicitar os requisitos. Estes são apenas alguns dos vários problemas que podem surgir durante o processo.

4.1.1 Brainstorming

Permite o surgimento e o refinamento de ideias, é importante que todos participem ativamente e que encorajem/enriqueçam as ideias dos colegas. Ideias sem relação entre si, quando combinadas podem resultar em soluções criativas para um dado problema. Ao final do brainstorming as ideias que agregam ao produto são revisadas, analisadas e priorizadas.

4.1.2 Entrevista

Tradicional, simples e poderosa, assim pode ser descrita esta técnica. É uma importante técnica na elicitação de requisitos, pois é uma técnica que pode ser aplicada praticamente em qualquer contexto e pode trazer bons resultados no início do processo. Apesar de ser uma técnica simples, é importante que seja feito um planejamento para a entrevista a fim de evitar a perda de foco e também é essencial para que a reunião não se estenda além do necessário.

O contexto é de extrema importância em uma entrevista, uma vez que isso vai possibilitar um maior aproveitamento na dissertação dos assuntos a serem tratados na entrevista. O entrevistador deve estar disposto a escutar e possivelmente mudar de opinião durante a entrevista, porém isso deve ser feito de forma que a produtividade e o foco sejam mantidos.

Ao final da entrevista é importante fazer uma validação de tudo aquilo que foi documentado, verificando assim se a informação adquirida corresponde às necessidades do cliente.

4.1.3 Workshop

É uma técnica de elicitação estruturada, realizada por um grupo composto por stakeholders escolhidos para representar a organização, por um facilitador(geralmente um membro neutro) e pela equipe de analistas. Diferentemente de uma reunião, o workshop tem como objetivo promover o trabalho em equipe e a interação entre os membros. Dentro dessa técnica podem ser utilizadas outras técnicas, como por exemplo o brainstorming, no final são produzidos documentos que contemplam os requisitos e as decisões acordadas a respeito do que será desenvolvido.

5 Gerenciamento de requisitos

5.1 Atributos de um requisito

Os atributos contêm as informações mais importantes daquele requisito, são suas propriedades, e é a partir dessas propriedades que se tem informações como: data de criação, origem, importância, dentre outras. Demandam atenção durante a sua especificação, pois fornecem importantes informações sobre o estado do projeto, além de permitirem que sejam realizadas consultas dentro de todos os requisitos. Os atributos definidos para gerenciar os requisitos durante a execução do nosso projeto foram:

5.1.1 Prioridade

- **Alta** - Indica vital importância para o sistema e para os stakeholders, normalmente agrega um grande valor aos usuários.
- **Média** - Indica relativa importância para o sistema e para os stakeholders, normalmente são requisitos que não foram compreendidos em sua totalidade pela equipe e podem ser implementados quando o cliente achar necessário.
- **Baixa** - Indica uma baixa importância para o sistema e para os stakeholders, normalmente representam funcionalidades opcionais para a realização do sistema.

5.1.2 Complexidade

- **Alta** - Indica alto nível de dificuldade, demandando um alto nível de esforço para a sua implementação.
- **Média** - Indica um relativo nível de dificuldade, demandando um moderado nível de esforço para a sua implementação.
- **Baixa** - Indica um baixo nível de dificuldade, demandando um baixo nível de esforço para a sua implementação.

5.1.3 Responsável

Identifica na equipe o membro responsável pelo requisito naquele momento.

5.1.4 Status

- **Finalizado** - Indica a realização daquele requisito e sua implementação no sistema.
- **Em progresso** - Indica que o requisito está sendo implementado no sistema.
- **Não iniciado** - Indica que o processo de implementação daquele requisito não teve início.

5.2 Rastreabilidade de requisitos

Rastreabilidade de software é a habilidade de relacionar artefatos criados durante o ciclo de vida de desenvolvimento de um sistema de software.[1] Os elementos rastreáveis do projeto são chamados de itens de rastreabilidade. O rastreamento de requisitos pode ser visto como uma habilidade de acompanhar e descrever a vida de um requisito em abas as direções. A pré-rastreabilidade documenta o contexto a partir do qual emergem os requisitos, já a pós-rastreabilidade vincula os requisitos ao desenho do sistema e sua implementação.(DAVIS, 1993). (FINALIZAR)

5.3 Gestão de requisitos