

**CENTRO UNIVERSITÁRIO DE ANÁPOLIS – UniEVANGÉLICA**  
**CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO (BEC)**

JANAINA LIMA GUIMARÃES

**AVALIAÇÃO DE CASOS DE USO E MÉTODO KNIGHT PARA DOCUMENTAÇÃO  
DE REQUISITOS EM PROJETOS ÁGEIS**

ANÁPOLIS-GO

2017

JANAINA LIMA GUIMARÃES

**AVALIAÇÃO DE CASOS DE USO E MÉTODO KNIGHT PARA DOCUMENTAÇÃO  
DE REQUISITOS EM PROJETOS ÁGEIS**

Pesquisa apresentada ao Curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis - UniEVANGÉLICA como requisito parcial à aprovação na disciplina Trabalho de Conclusão de Curso II sob orientação do Prof. Kleber Silvestre Diogo.

ANAPOLIS-GO

2017

JANAINA LIMA GUIMARÃES

**AVALIAÇÃO DE CASOS DE USO E MÉTODO KNIGHT PARA DOCUMENTAÇÃO  
DE REQUISITOS EM PROJETOS ÁGEIS**

Pesquisa apresentada ao Curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis - UniEVANGÉLICA como requisito parcial à aprovação na disciplina Trabalho de Conclusão de Curso II sob orientação da Prof. Kleber Silvestre Diogo.

Banca Examinadora

.....  
Prof. Kleber Silvestre Diogo  
Orientador

.....  
Prof<sup>a</sup>. Raissa dos Santos Vieira  
Convidado

.....  
Prof<sup>a</sup>. Viviane Carla Batista Pocivi  
Convidado

Nota:.....

Anápolis, ..... de .....de 2017.

À minha família, àqueles que me apoiaram, me deram dicas e compartilharam seu conhecimento, em especial ao Prof. Kleber Silvestre Diogo pelas orientações e correções e a Deus pois sem Ele nada disso seria possível.

## LISTA DE FIGURAS

Figura 1 – Metodologia do Trabalho.....	15
Figura 2 - Um caso de uso de solicitação de férias.....	20
Figura 3 - Elementos da notação BPMN.....	23
Figura 4 - Processo de aprovação de férias .....	24
Figura 5 - Questões respondidas pelo modelo de negócio.....	25
Figura 6 – Método Knight.....	26
Figura 7 – Scrum.....	32
Figura 8 - OpenUP.....	34
Figura 9 – Kanban.....	35
Figura 10 - Processo do sistema acadêmico Virtoo .....	40
Figura 11 - Modal de detalhes da tarefa .....	41
Gráfico 01 - Modelos e sua aderência às recomendações da IEEE.....	43
Gráfico 02 - Comparativo de respostas pela função do entrevistado .....	44
Gráfico 03 - Comparando a documentação com BPMN e a documentação padrão com relação às características recomendadas pela IEEE.....	45
Gráfico 04 - Principais problemas no levantamento de requisitos.....	46
Gráfico 05 – Total de problemas reportados de acordo com a função do entrevistado.....	47
Gráfico 06 - Problemas reportados, de acordo com a função do entrevistado.....	47
Figura 12 – Modal de detalhes de um caso de uso – modelo anterior .....	51
Figura 13 – Caso de uso Manter Mantenedora, modelado de acordo com a proposta.....	51
Figura 14 - Fluxo principal do sistema Virtoo .....	57
Figura 15 - Processo da Fábrica de Tecnologias Turing.....	38

## LISTA DE TABELAS

Tabela 1 - Cronograma Programático do TCC I.....	37
Tabela 2 - Elementos BPMN e significado no modelo proposto.....	50

## LISTA DE SIGLAS

FTT	Fábrica de Tecnologias Turing
BPMN	<i>Business Process Model and Notation</i>
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IEEE-830	Norma da IEEE que descreve o conteúdo e a qualidade de uma boa especificação de exigências de software.
EES	Especificação de exigências de software
OMG	<i>Object Management Group</i>
CMMI	<i>Capability Maturity Model Integration</i>
PMI	<i>Project Management Institute</i>
RH	Recursos Humanos
NCE	Núcleo de Computação e Eletrônica da Universidade Federal do Rio de Janeiro.
PO	<i>Product owner</i>
IBM	<i>International Business Machines</i>
ISPVIDA	Instituto Superior Politecnico Vida Isp-vida Lubango
ISTEL	Instituto Superior De Teologia Evangelica No Lubango
HTML	<i>HyperText Markup Language</i>

# Sumário

LISTA DE FIGURAS.....	v
LISTA DE TABELAS .....	vi
LISTA DE SIGLAS.....	vii
RESUMO.....	x
ABSTRACT .....	xi
1 INTRODUÇÃO .....	11
2 REFERENCIAL TEÓRICO .....	16
2.1 Engenharia de <i>Software</i> .....	16
2.2 Requisitos de Software .....	17
2.3 Norma IEEE-830 .....	18
2.4 Modelo de Caso de uso ( <i>use-case</i> ).....	20
2.5 Processo de negócio.....	21
2.5.1 BPMN (Business Process Modeling Notation).....	22
2.6 Método Knight .....	24
2.6.1 Fase 1 – Visão Geral do Problema.....	26
2.6.2 Fase 2 - Visão geral da solução .....	28
2.7 Scrum.....	31
2.8 OpenUP .....	33
2.9 Kanban.....	35
2.10 Fábrica de Tecnologias Turing.....	35
2.10.1 Projeto Virtoo.....	37
3 RESULTADOS .....	38
3.1 Necessidades na elicitação e documentação dos requisitos .....	38
3.2 Etapas para a aplicação e avaliação de documentação com BPMN .....	38
3.3 Proposta de nova forma de documentação.....	39
3.4 Comparação dos modelos.....	41
3.4.1 Questionário e entrevistas .....	42
3.4.2 Dados obtidos.....	43
3.5 Pontos de melhoria identificados.....	48
3.6 Proposta de melhoria .....	48
4 Conclusão .....	53
4.1 Trabalhos futuros .....	53
REFERÊNCIAS BIBLIOGRÁFICAS .....	54

APÊNDICES.....	56
APÊNDICE A - Questionário.....	56
Anexos.....	57
Documento do projeto Virtoo com modelagem de processo.....	57
Processo da Fábrica de Tecnologias Turing.....	58

## RESUMO

Observando a utilização do modelo de documentação com caso de uso aplicado na Fábrica de Tecnologias Turing (FTT)<sup>1</sup>, percebeu-se que a documentação dos requisitos utilizada gerava burocracia para atualização dos requisitos, já que era necessário atualizar diversos documentos e por esse motivo não era possível acompanhar as mudanças dos requisitos, além da dificuldade no entendimento do projeto e seu contexto de aplicação na organização do cliente. Após a realização do *brainstorming* realizado com os analistas de requisitos, parte dos analistas de teste e desenvolvedores que participavam do projeto, decidiu-se comparar o modelo de documentação já aplicado na FTT, o modelo de caso de uso, com o Método Knight para documentação de requisitos baseados no modelo de negócio, afim de gerar um novo modelo aderente à realidade do projeto Virtoo. O objetivo da pesquisa consiste em comparar o uso de técnicas de elicitação e documentação de requisitos em projetos ágeis na FTT e gerar um modelo de documentação de requisitos utilizando BPMN<sup>2</sup>. Para alcançar o objetivo proposto foi realizada uma análise do cenário do projeto utilizando o modelo de caso de uso, um estudo e adaptação do Método Knight para ser aplicado na FTT, aplicação do método de documentação adaptado e análise do impacto do novo método adotado. A partir desta pesquisa foi possível identificar o desempenho do modelo proposto, que conseguiu ser mais aderente às normas da IEEE-830<sup>3</sup> com relação ao modelo utilizado no FTT anteriormente, facilitou e centralizou a busca por informações do caso de uso, melhorou o entendimento do contexto da aplicação e o comportamento esperado. Como proposta para trabalhos futuros recomenda-se a aplicação de treinamentos sobre BPMN para os novos integrantes do projeto, bem como a modelagem do fluxo também no nível do caso de uso.

**Palavras-chave:** Fábrica de software. Elicitação de requisitos. Documentação de requisitos. Modelagem de processo. Método Knight. Caso de Uso.

---

<sup>1</sup> A FTT é uma unidade do curso de Engenharia de Computação da UniEvangélica, onde os alunos podem vivenciar experiências de desenvolvimento de projetos reais com o acompanhamento do corpo docente do curso.

<sup>2</sup> *Business Process Model and Notation* (BPMN), é uma notação da metodologia de gerenciamento de processos de negócio e trata-se de uma série de ícones padrões para o desenho de processos.

<sup>3</sup> A norma da IEEE que descreve as abordagens recomendadas para a especificação de requisitos de software e é utilizada como métrica de avaliação nesta pesquisa.

## ABSTRACT

Observing the use of the use case documentation applied at the Turing Software House (FTT), it was noticed that the requirements documentation used increased the bureaucracy to update the requirements, since it was necessary to update several documents and for that reason it was not possible to keep up with changes in requirements, as well as the difficulty to understanding the project and its application context in the client's organization. After brainstorming with the requirements analysts, part of the test analysts and developers that participated in the project, it was decided to compare the documentation model applied in the FTT, the use case model, with the Knight Method for documentation of requirements based on the business process model, in order to generate a new model adherent to the reality of the Virtoo project. The objective of this research is to compare the use of elicitation techniques and requirements documentation in agile projects in FTT and to generate a requirements documentation model using BPMN<sup>4</sup>. In order to reach the proposed objective, a project scenario analysis was performed using the use case model, a study and adaptation of the Knight Method to be applied in the FTT, the application of the adapted documentation method and analysis of the impact of the new method adopted was made. From this research it was possible to identify the performance of the proposed model, which managed to be more adherent to the IEEE-830 standards in relation to the use case model, facilitated and centralized the search for use case information, improved understanding of the context of the application and the software's expected behavior. As a proposal to future work it's recommended to apply a BPMN training to the new project members, as well as modeling the process flow also at the use case level.

**Keywords:** Software House. Elicitation of requirements. Documentation of requirements. Process modeling. Knight method. Use case.

---

<sup>4</sup> *Business Process Model and Notation* (BPMN), is a notation of process and business management methodology and it is a series of standard icons for process design.

## 1 INTRODUÇÃO

Os projetos ágeis são caracterizados por sua capacidade de adaptação às mudanças, principalmente de requisitos, e por realizarem o desenvolvimento em ciclos, contendo as etapas de elicitação de requisitos, desenvolvimento e testes. “Um dos problemas básicos da engenharia de *software* é o levantamento e documentação dos requisitos dos produtos de *software*” (FILHO,2013, p.8). Uma elicitação de requisitos incompleta ou incorreta, impacta em todo o ciclo e leva o projeto ao fracasso.

O objeto de estudo escolhido nesta pesquisa foi o projeto Virtoo<sup>5</sup>, realizado na FTT a partir do segundo semestre de 2016. A FTT é uma unidade do curso de Engenharia de Computação da UniEvangélica, onde os alunos podem vivenciar experiências de desenvolvimento de projetos reais com o acompanhamento do corpo docente do curso. A FTT possui uma estrutura para atender vários projetos simultaneamente, com equipes multidisciplinares para cada projeto, mas à época do desenvolvimento desta pesquisa, a estrutura havia sido ajustada para atender somente ao projeto Virtoo, devido à sua complexidade, passando a possuir equipes especializadas que trabalhavam em conjunto na construção do *software*.

No projeto Virtoo, uma das dificuldades encontradas foi que a forma como o processo de elicitação e documentação dos requisitos funcionava gerava burocracia para atualização dos requisitos, devido ao número de documentos a serem editados além de exigir maior tempo para a adequação das mudanças dos requisitos do projeto. Estes sofriam ajustes frequentes devido à complexidade do projeto e à dificuldade no entendimento dos requisitos e do contexto da aplicação do *software*, por se tratar de um projeto internacional.

Nesse contexto, como é possível otimizar a elicitação e especificação de requisitos de modo a facilitar o seu entendimento e manutenibilidade ao longo do processo de desenvolvimento de *software* em projetos ágeis?

O escopo deste trabalho abrange a elicitação e especificação de requisitos, etapas que fazem parte da engenharia de requisitos, uma das disciplinas da engenharia de *software*. “Uma especificação de requisitos é um acordo entre o usuário final e o desenvolvedor de sistemas” (GHEZZI; JAZAYERI; MANDRIOLI,

---

<sup>5</sup> Sistema acadêmico e financeiro para uma universidade de angola.

2002.p.161, tradução nossa), e serve para guiar o desenvolvimento e teste do produto de *software*.

O sucesso de um projeto de *software* está, em grande parte, relacionado a uma boa elicitación de requisitos. É pouco provável que um sistema, mesmo que sem erros, seja utilizado pelo usuário final se não atender aos requisitos mínimos, visto que “projetos e codificação perfeitos são de pouca utilidade quando existem erros nos requisitos” (MACHADO,2011, p.40). Um projeto não falha devido a somente um fator, mas “um dos problemas básicos da engenharia de *software* é o levantamento e documentação dos requisitos dos produtos de *software*” (FILHO,2013, p.8). Isso acontece devido a vários problemas que ocorrem nessa etapa do projeto. Machado (2011) cita que o entendimento incompleto das necessidades, problemas de comunicação entre os analistas e usuários, omissão de requisitos e a volatilidade dos requisitos, devido a fatores internos e externos são alguns dos problemas que afetam a elicitación de requisitos.

Pressman (2016) afirma que “o modo de especificação se reflete na qualidade da solução”. Existem diversas abordagens para a elicitación e especificação dos requisitos com o intuito de amenizar o impacto dos problemas citados anteriormente, dentre elas estão: Casos de uso, User Story, método Mac Knight, prototipagem de requisitos, oficinas de requisitos, dentre outras.

Observando a utilização do modelo de documentação com caso de uso aplicado na FTT percebeu-se que havia muita burocracia ao documentar os requisitos. A documentação não atendia a um dos princípios do manifesto ágil (aceitar mudanças de requisitos) e isso dificultava a adequação dos requisitos de acordo com as necessidades da organização. Outra necessidade encontrada foi melhorar o entendimento que a equipe tinha em relação ao propósito do sistema e sua aplicação na organização, já que se tratava de um projeto internacional, desenvolvido à distância. Neste trabalho foi realizada uma comparação entre a utilização da documentação com BPMN e com casos de uso afim de verificar qual delas se adequava melhor ao contexto do projeto Virtoo. O modelo de caso de uso foi selecionado pois já era utilizado na FTT, além de ser amplamente difundido no mercado. O Método Knight foi selecionado pois propõe a utilização do modelo de negócio como base para a elicitación e documentação dos requisitos e uma das

necessidades encontradas foi a dificuldade de entendimento do modelo de negócio e qual o cenário no qual o sistema construído seria inserido.

Sobre o método de Mac Knight para elicitação de requisitos a partir do modelo de negócios, “um aspecto que recebe especial atenção do método é a colaboração entre os analistas de sistemas e pessoas da organização, lembrando muito este ponto do manifesto ágil” (MACHADO,2011, p.163), o que possibilita minimizar os problemas de comunicação entre analistas e usuários. Mais importante do que a notação ou frameworks é: Saber identificar os processos de negócio, os dados consumidos e gerados por ele e saber detalhá-los sem a interferência de tecnologia (MACHADO,2011).

A partir deste estudo será possível avaliar os resultados da aplicação do modelo de documentação com BPMN e gerar um modelo de documentação de requisitos baseado em processo para ser aplicado em projetos ágeis.

O objetivo da pesquisa consiste em comparar o uso de técnicas de elicitação e documentação de requisitos em projetos ágeis na FTT e gerar um modelo de documentação de requisitos utilizando BPMN<sup>6</sup>.

Para isso, os objetivos específicos abaixo devem ser realizados:

1. Identificar técnicas de elicitação e documentação atualmente aplicadas na FTT e os principais problemas gerados pela sua utilização.
2. Discutir com os envolvidos no projeto Virtoo as técnicas de elicitação e documentação de requisitos adequadas para o projeto.
3. Selecionar a técnica que melhor se adapte ao contexto da FTT e seja capaz de solucionar ou minimizar os problemas encontrados.
4. Adaptar o modelo de documentação de acordo com a técnica escolhida
5. Aplicar o modelo selecionado em um projeto da FTT.
6. Medir os resultados da aplicação de acordo com os requisitos de uma boa especificação de software definidos pela IEEE-830, comparando a utilização do modelo adaptada com o uso do modelo anterior.
7. Aplicar ações corretivas no modelo.

---

<sup>6</sup> *Business Process Model and Notation* (BPMN), é uma notação da metodologia de gerenciamento de processos de negócio e trata-se de uma série de ícones padrões para o desenho de processos.

Para a realização desta pesquisa foi estudada documentação de requisitos com Casos de Uso, por ser a técnica utilizada na FTT. Foi realizada uma avaliação dos resultados das *sprints* anteriores com base nos relatos das reuniões de retrospectiva e na realização de um *brainstorming* entre os integrantes da equipe de analistas do projeto Virtoo para listar e discutir os problemas e elencar as opções disponíveis para solucionar ou, ao menos, minimizar os problemas encontrados.

Dentre as opções discutidas, foi selecionada a documentação a partir do modelo de negócio, que tem por base o Método Knight para a elicitación dos requisitos de *software*, visto que um dos principais problemas encontrados estava relacionado à definição do que o *software* deveria fazer para se adequar ao contexto da empresa solicitante.

Foi desenvolvido um modelo dinâmico de elicitación e especificação de requisitos com *Business Process Model and Notation* (BPMN) utilizando a ferramenta Bizagi, que foi aplicado na FTT por um semestre. Os resultados da aplicação foram medidos e avaliados a partir de observação do novo contexto do projeto, entrevistas e de questionários baseados nas características de um bom documento de Especificação, segundo a IEEE-830<sup>7</sup>, sendo: Correto; Não ambíguo; Completo; Consistente; Classificável por importância e/ou estabilidade; Verificável; Modificável; Rastreável.

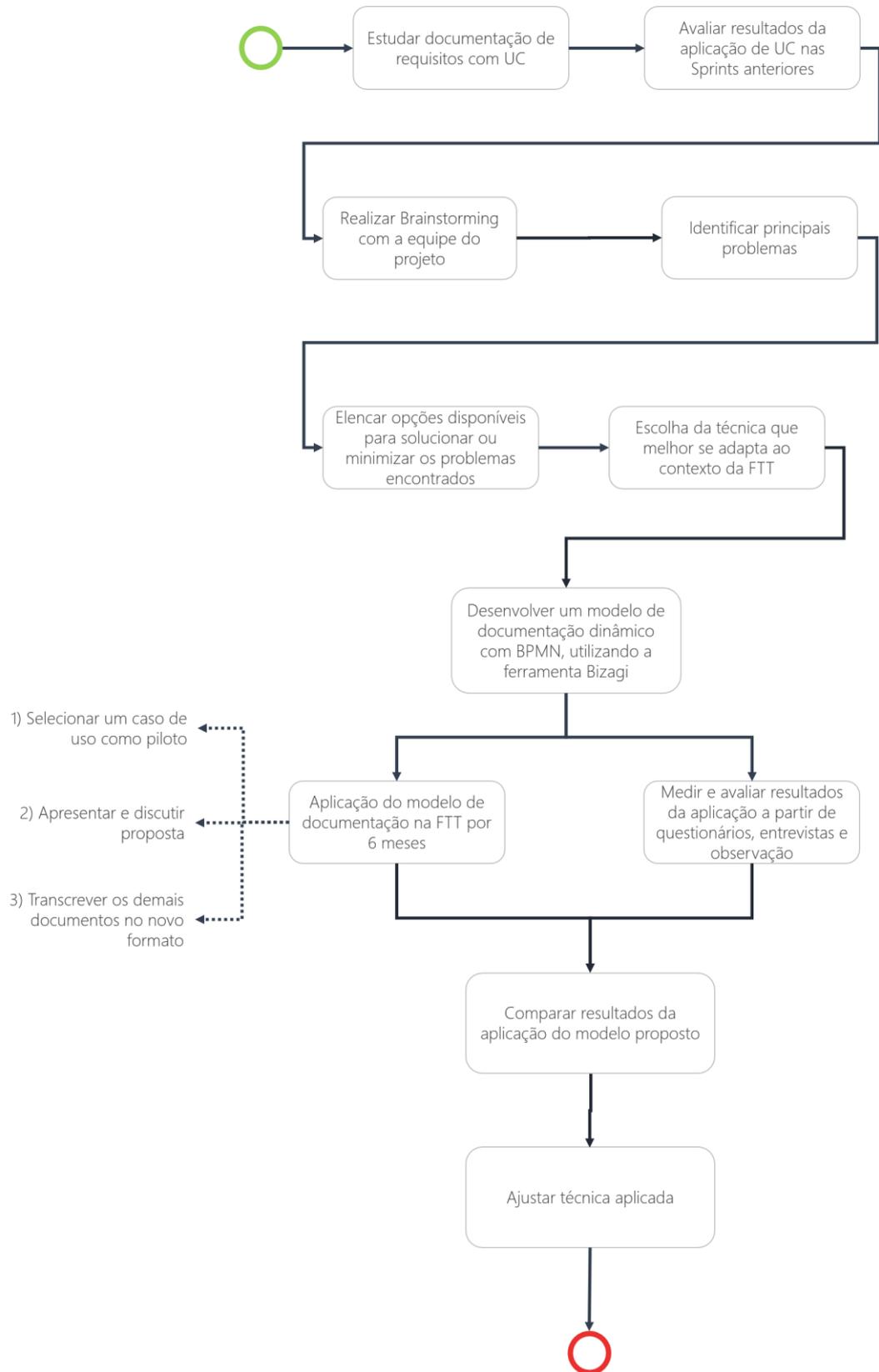
Os resultados alcançados a partir da utilização das duas técnicas foram comparados avaliando a aderência às normas da IEEE-830 após a aplicação. Melhorias para trabalhos futuros foram identificadas e estão descritas no item 3.6 desta pesquisa.

A Figura 1 ilustra a metodologia proposta para este trabalho.

---

<sup>7</sup> Norma da IEEE que descreve as abordagens recomendadas para a especificação de requisitos de software

Figura 1 – Metodologia do Trabalho



Fonte: Autores, 2017

## 2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados assuntos essenciais para a compreensão do trabalho como um todo. Os tópicos abaixo relacionados à Engenharia de *Software*, Requisitos de *software*, *norma IEEE-830* e modelo de caso de uso apresentam informações relacionadas ao desenvolvimento de projetos de *software*. Na sessão Processo de Negócio será apresentada a notação *Business Process Model and Notation* (BPMN) e logo em seguida o Método Knight para elicitação de requisitos a partir do modelo de negócio será apresentado e detalhado.

Os itens Scrum, OpenUp e Kanban apresentam a definição, respectivamente, da metodologia de gestão de projetos, do processo de desenvolvimento do *software* e do quadro de acompanhamento de atividades da FTT, que também será apresentada neste capítulo.

### 2.1 Engenharia de *Software*

Antes de falar sobre a engenharia de *software* é importante definir o que é o *software*. Algumas pessoas definem “*software*” como sendo programas de computador. Porém “*software* não é apenas o programa, mas também todos os dados de documentação e configuração associados, necessários para que o programa opere corretamente” (SOMMERVILLE,2007. p.4). Pressman (2016) define *software* da seguinte forma:

“Uma descrição de *software* num livro didático poderia assumir a seguinte forma: “*Software* é: (1) instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados; (2) estruturas de dados que possibilitam que os programas manipulem adequadamente a informação; e (3) documentos que descrevem a operação e o uso dos programas”

A engenharia de *software* “é uma disciplina de engenharia relacionada com todos os aspectos da produção de *software*, desde os estágios iniciais de especificação do sistema até a sua manutenção” (SOMMERVILLE,2007. p.5).

Existem quatro processos fundamentais comuns a todos os processos de *software*, sendo eles:

- a) Especificação de *software*: referente à definição das funcionalidades e restrições do software a ser desenvolvido.
- b) Desenvolvimento de *software*: produção do *software*, de modo que atenda às suas especificações.
- c) Validação do *software*: Etapa na qual o *software* deve ser validado para garantir que ele faz o que o cliente deseja.
- d) Evolução do *software*: Referente às etapas de manutenção e evolução, na qual o *software* deve evoluir para atender às necessidades mutáveis do cliente (SOMMERVILLE, 2007).

A engenharia de *software* abrange métodos, procedimentos e ferramentas que possibilitam ao gerente o controle do processo de desenvolvimento do *software* e oferece ao profissional uma base para construir um *software* de alta qualidade (PRESSMAN, 2016).

## 2.2 Requisitos de Software

De acordo com a literatura “a disciplina de requisitos reúne as atividades que visam obter o enunciado completo, claro e preciso dos requisitos de um produto de *software*” (FILHO, 2013).

Machado (2011) define requisitos de *software* de várias formas: (1) uma condição ou capacidade com a qual o sistema deve estar em conformidade; (2) uma especificação do que deve ser implementado; (3) uma restrição que deve ser atendida; (4) uma condição ou capacidade necessária para o usuário resolver um problema ou alcançar um objetivo; (5) uma condição ou capacidade para satisfazer um contrato, padrão ou especificação; (6) uma documentação de uma condição ou capacidade.

“A análise de requisitos é usualmente a primeira fase do desenvolvimento de *software* em larga escala” (GHEZZI; JAZAYERI; MANDRIOLI, 2002.p.6, tradução

nossa). Requisitos são uma forma de expressar características e restrições do produto de *software* que independe da tecnologia utilizada. A etapa de análise de requisitos é uma das partes mais críticas e propensas a erros no desenvolvimento de *software* (MACHADO, 2011).

Um dos fatores que impacta na qualidade dos *softwares* desenvolvidos atualmente é a deficiência na formação de analistas de sistemas. Embora, a cada dia que se passa, as certificações CMMI, PMI, entre outras, fiquem mais populares, ainda existe um déficit no ensino sobre processos de negócios e como se aplicam sistemas automatizados a esses processos. Isso faz com que boa parte dos analistas de sistemas saibam, por exemplo, criar casos de uso e *User stores*, mas somente como uma forma de especificar o requisito que deve ser programado, mas não entendem o que ele significa no contexto do negócio do cliente (MACHADO, 2011).

Alguns paradigmas sugerem soluções para este problema, como é o caso do método Knight que será apresentado mais à frente.

Uma importante questão que surge quando um novo projeto precisa ser construído é como caracterizar os requisitos corretamente, visto que “é preciso especificar e modelar o problema de maneira que seja possível criar um projeto efetivo” (MACHADO, 2011. p.179). Dentre os modelos de especificação de requisitos, serão apresentados o Casos de Uso e o método Knight para a elicitação e especificação dos requisitos.

### **2.3 Norma IEEE-830**

A norma IEE-830 descreve o conteúdo e a qualidade de uma boa especificação de exigências de software (EES), aqui chamada de especificação de requisitos de software ou somente especificação de requisitos, apresenta referencias, informações base a serem consideradas durante a escrita da especificação de requisitos, as partes de um documento de especificação e exemplos de possíveis tipos de estruturas de documentos EES.

Segundo (GONÇALVES et al., 2004) um documento de especificação de requisitos deve ser:

- a. Correto;**

Se e só se, todas as exigências expressas nele forem correspondidas pelo software

**b. Não ambíguo;**

Se e somente se todas as exigências expressas nele têm apenas uma única interpretação

**c. Completo;**

Um documento de especificação é considerado *completo* se inclui os seguintes elementos:

- Todas as exigências significantes estão reconhecidas e tratadas.
- Estão especificadas as respostas do software a todos os tipos de entradas de dados em todas as classes de situações.
- Legendas e referências completas para todas as figuras, tabelas e diagramas do documento de especificação bem como a definição de todos os termos e unidades de medida.

**d. Consistente;**

Se e somente se, nenhum sub-conjunto individual de exigências descrito nele entra em conflito.

**e. Classificável por importância e/ou estabilidade;**

Se cada exigência nele contido tem associada um identificador de estabilidade e/ou importância.

**f. Verificável;**

Se, e somente se, cada exigência especificada é verificável, ou seja, existe um processo finito e de custo aceitável através do qual uma pessoa ou uma máquina pode verificar que o produto de software cumpre essa exigência.

**g. Modificável;**

Se, e somente se, a sua estrutura e estilo permitem que as mudanças a exigências sejam realizadas de forma fácil, completa e consistente, preservando simultaneamente estrutura e estilo.

**h. Rastreável;**

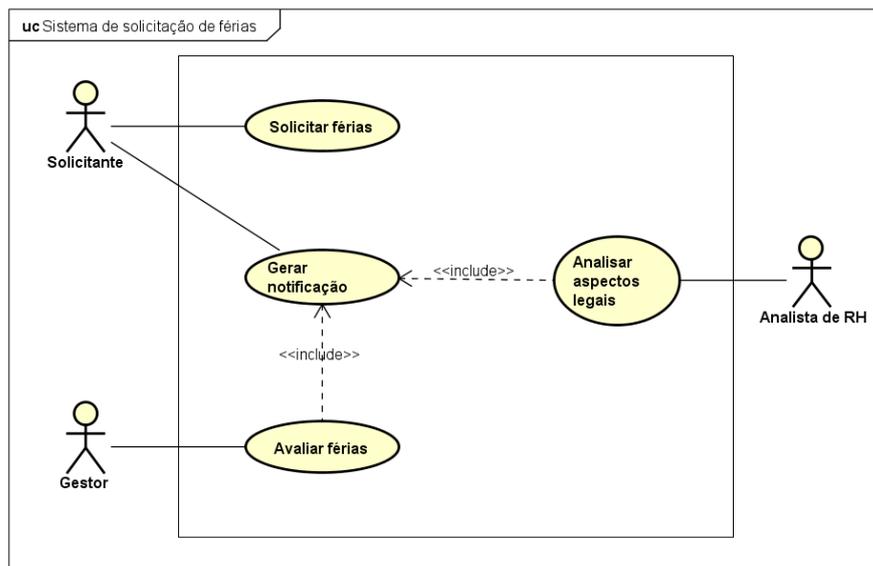
Uma especificação é rastreável se cada uma das suas exigências é clara e facilitadora da identificação da mesma exigência em versões futuras do desenvolvimento ou da documentação.

## 2.4 Modelo de Caso de uso (*use-case*)

Um caso de uso especifica o comportamento do *software* a partir de uma perspectiva externa, consiste na interação entre o ator e a máquina e é uma sequência de ações que geram um resultado observável pelo ator (MACHADO, 2011).

“Os casos de uso constituem uma técnica baseada em cenários para eliciação de requisitos [...]. Eles se tornaram uma característica fundamental da notação UML (linguagem de modelagem unificada) para descrever modelos de sistemas orientados a objetos.” (SOMMERVILLE, 2007. p. 102). “Em sua forma mais simples, um caso de uso identifica o tipo da interação e agentes envolvidos”. (SOMMERVILLE, 2007, p. 102). Para exemplificar o uso de um caso de uso, imagine um sistema de solicitação de férias, no qual o solicitante/funcionário deve ter a opção de solicitar as férias e ser notificado quando ela for aprovada ou reprovada tanto pelo RH quanto pelo seu gestor. O analista de RH deve ter a opção de verificar os aspectos legais e informar se o funcionário solicitante tem direito às férias ou não. Como última funcionalidade, o gestor deve avaliar a solicitação de férias, liberando ou não o funcionário.

Figura 2 - Um caso de uso de solicitação de férias



powered by Astah

Fonte: Autores, 2017

Após a identificação dos requisitos estes “devem ser especificados em detalhe e documentados” (MACHADO, 2011. p.260). Para a documentação de requisitos utilizando a modelagem de casos de uso, Machado (2011) cita que para cada caso de uso do modelo são descritos os seguintes aspectos:

- a) Descrição breve: um parágrafo descrevendo o caso de uso de forma sucinta.
- b) Fluxo de eventos básico: Descreve a interação entre ator e sistema. Os eventos devem ser descritos de forma independente da solução técnica. O fluxo básico é o fluxo que ocorre com mais frequência, e não contempla tratamento de erros ou alternativas.
- c) Fluxo de eventos alternativos: descreve os eventos alternativos. Normalmente trata de ações opcionais relacionados ao caso de uso.
- d) Fluxo de eventos de exceção: referente ao tratamento de erros e exceções que podem acontecer no caso de uso.
- e) Regras de negócio: são regras de negócio muito complexas, requisitos não funcionais ou detalhes que não são adequados para a inserção nos fluxos, porém são importantes para o *software*.
- f) Protótipo de interface com o usuário: é uma sessão opcional, que não aparece no modelo clássico de especificação de caso de uso, tem por objetivo facilitar o entendimento do fluxo de eventos e “não representam necessariamente todos os detalhes da tela final do sistema” (MACHADO, 2011. p.261)

## 2.5 Processo de negócio

Os processos fazem parte das atividades diárias tanto dos colaboradores quanto da própria gestão da organização (BRITTO, 2011) e são constituídos de uma sequência de atividades, formada por tarefas que têm o objetivo de gerar um resultado que surpreenda o cliente e que “tudo o que realizamos dentro de uma organização está relacionado a um processo, independentemente de sua complexidade” (ARAUJO, GARCIA e MARTINES, 2011. p.29).

Atualmente existem muitas técnicas de modelagem de processos, algumas mais complexas, outras mais simples, umas mais utilizadas e outras menos conhecidas. Dentre as principais técnicas, escolhemos o *Business Process Model*

*and Notation* (BPMN) para ser utilizado neste trabalho por ser uma modelagem bem difundida e de fácil entendimento.

### **2.5.1 BPMN (*Business Process Modeling Notation*)**

O BPMN, Notação de Modelo de Processo de Negócio, é um padrão de notação desenvolvido pelo *Business Process Management Initiative* (BPMI) e publicado em 2004. No ano seguinte, o BPMI se fundiu com a *Object Management Group* (OMG), “uma associação internacional de padrões de tecnologia, aberta, sem fins lucrativo, fundada em 1989” (OBJECT MANAGEMENT GROUP, 2017, tradução nossa).

A criação da notação tem dois objetivos principais:

1. Servir de elo entre o desenho e a implementação do processo.
2. Ser intuitivo e de fácil entendimento para todos os públicos, sejam eles analistas de processo, clientes, técnicos entre outros.

Um processo pode representar um conjunto de atividades pertencentes a uma ou diversas organizações, sendo possível mapear três tipos de processos:

- Processos internos: que são realizados dentro da organização
- Processos abstratos: Pertencentes a entidades externas, nos quais seu detalhamento não é mapeado.
- Processos de colaboração: Também não pertencem à organização, mas que devido à sua importância, são detalhados da mesma forma que os processos internos (ARAUJO, GARCIA e MARTINES, 2011).

Os objetos de fluxo são os principais elementos gráficos e definem o comportamento dos processos de negócio. São eles: eventos, atividades e gatilhos, representados na Figura 4.

Figura 3 - Elementos da notação BPMN



Fonte: <http://www.bpmn.org/> , s.d.

**Evento:** pode iniciar o processo, mudar seu caminho ou concluir um processo. É um acontecimento real que provoca uma ação e está dividido em três tipos: De Início, Intermediários e de Fim (BRITTO,2011).

**Atividade:** é uma “decomposição do trabalho em ações a serem realizadas dentro do processo” (BRITTO,2011. p. 70). Normalmente são independentes, agregam valor e geram produtos mensuráveis.

**Gatilho:** chamado de regra por Britto, “é o entendimento e o estabelecimento formal de dependência entre as atividades do processo” (BRITTO,2011. p. 70). É normalmente usada para definir os próximos passos no processo.

Existem diversas ferramentas de modelagem de processo de negócio aderentes ao *Business Process Model and Notation* (BPMN) , das quais selecionou-se o Bizagi<sup>8</sup> por possuir uma interface intuitiva além de disponibilizar recursos de exportação e criação de documentos e páginas navegáveis a partir da modelagem.

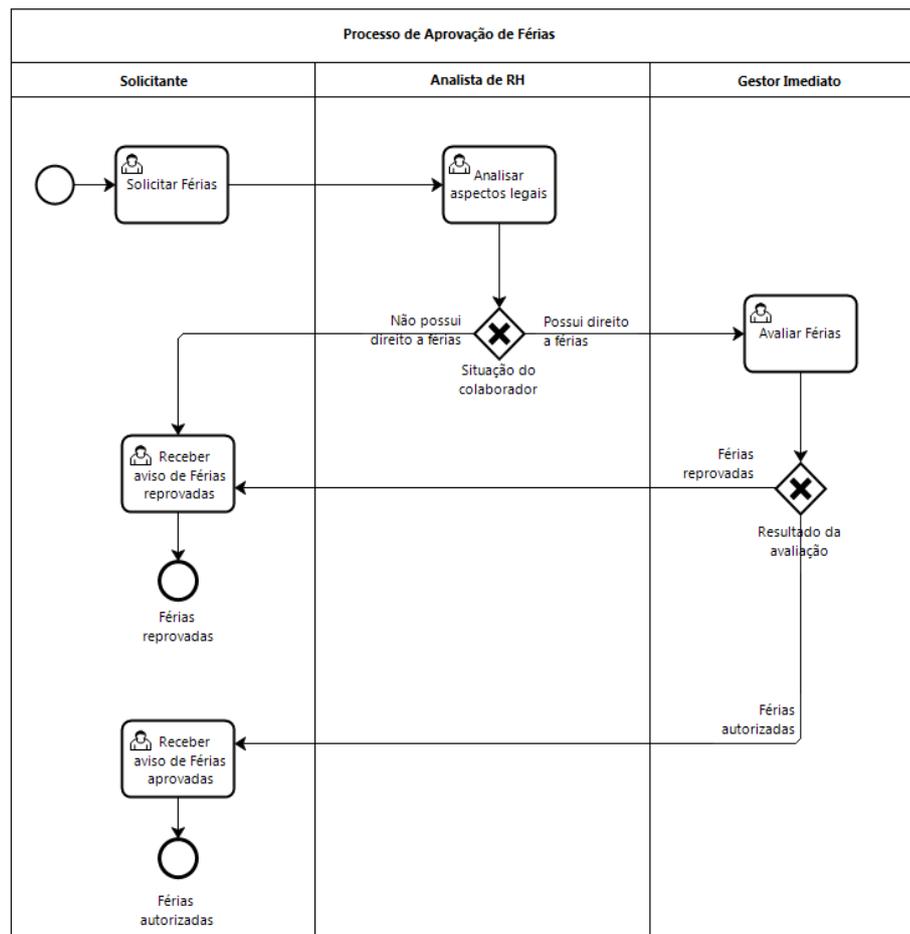
Para exemplificar o uso da notação *Business Process Model and Notation* (BPMN), a Figura 5 ilustra um exemplo de processo de aprovação de férias, sendo:

O processo inicia quando o solicitante realiza a solicitação de férias, que é enviada para a análise de aspectos legais, tarefa realizada pelo analista de RH. Caso o solicitante não possua direito a férias ele é notificado sobre a reprovação do

<sup>8</sup> Bizagi: É um software de gestão de processos de negócio que possui 3 módulos: Modeler, Studio e Engine, que possibilitam a modelagem, automação e distribuição/publicação do processo de negócio, respectivamente.

pedido de férias e o processo se encerra. Caso o solicitante possua direito a férias a solicitação deverá ser enviada ao seu gestor imediato para que este realize a avaliação de férias. Caso o gestor reprove a solicitação o processo se encerra, caso ele aprove o pedido de férias, o solicitante é notificado sobre a aprovação de suas férias e o processo se encerra.

Figura 4 – Processo de aprovação de férias



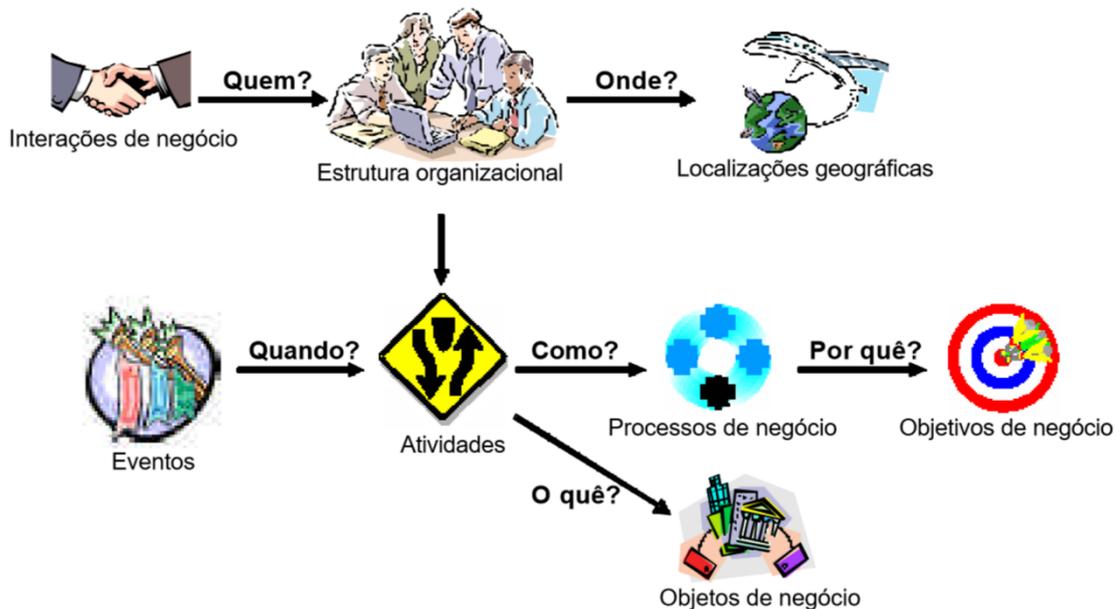
Fonte: <http://blog.iprocess.com.br/wp-content/uploads/2013/05/BPMN-pools-e-lanes-diagrama-vertical.png> (2013)

## 2.6 Método Knight

“O objetivo de um modelo de negócio é responder às seis perguntas clássicas sobre a organização: O que é feito? Quem faz? Quando? Onde? Por quê? Como?” (KNIGHT, 2004). A partir das respostas é possível obter uma visão ampla sobre a

organização e seu negócio. A Figura 6 mostra os conceitos presentes no modelo de negócio.

Figura 5 – Questões respondidas pelo modelo de negócio



Fonte: Knight (2004)

A partir da visualização do processo de negócio ganha-se subsídio para a análise de requisitos. Para a obtenção dos requisitos a partir do processo devemos identificar o que o sistema deve fazer para facilitar determinada atividade (TAKAY, 1998 apud MACHADO, 2011).

O Método Knight para a elicitación de requisitos a partir de modelo de negócio foi proposto por Debora Mac Knight, na sua tese de mestrado defendida em 2004 no Núcleo de Computação e Eletrônica (NCE) da Universidade Federal do Rio de Janeiro.

Segundo Machado (2011), a abordagem utiliza o modelo de negócio como base para identificar as principais necessidades que devem ser atendidas pelo *software* a ser desenvolvido. A partir desse entendimento, torna-se possível definir os requisitos do *software*. O objetivo do método é servir de guia durante a elicitación de requisitos, principalmente de *softwares* que tem como objetivo automatizar ou otimizar tarefas da organização. O método define como importante a interação entre

os analistas de sistemas e Stakeholders, através de reuniões para melhorarem o entendimento do problema e a proposta de uma solução. Também é levado em consideração a rastreabilidade dos requisitos, visto que cada funcionalidade está relacionada a uma necessidade do negócio, permitindo que, caso o negócio sofra alteração seja possível identificar as áreas do *software* que serão afetadas. O método proposto por Knight considera o modelo de negócio descrito em qualquer notação e na etapa final do método um documento que lista os requisitos do usuário é gerado. As demais etapas do processo de desenvolvimento de *software* não são contempladas no método.

O método Knight é dividido em duas etapas e cada etapa é composta por fases, conforme exibido na figura 7 adaptada de acordo com a especificação dada por Knight (2004).

Figura 7 – Método Knight



Fonte: Autores, 2017

### 2.6.1 Fase 1 – Visão Geral do Problema

Esta fase tem como objetivo “permitir que os engenheiros de *software* tenham uma visão do que é o negócio da organização e qual parte deverá ser considerada para atender à solicitação” (KNIGHT, 2004. p.66). Ela é composta pelas 3 primeiras etapas do método. Na primeira etapa, os processos são analisados afim de definir quais deles estão no escopo da solicitação de desenvolvimento de *software* e

permite que os engenheiros tenham uma visão geral e um melhor entendimento do negócio. Na segunda etapa, as necessidades que devem ser atendidas pelo sistema são identificadas. A terceira etapa consiste em identificar o impacto dessas necessidades, o que servirá como subsídio para a tomada de decisões no projeto.

#### **2.6.1.1 Etapa 1 – Identificar o contexto da solicitação**

Nesta etapa espera-se que o modelo de negócios já exista, pois é a partir dele que será definido o escopo do sistema. O primeiro passo é analisar o modelo de negócio para identificar quais os processos e sub processos estão envolvidos com sistema a ser construído, a partir da sua descrição. No segundo passo as atividades dos processos identificados são analisadas, afim de encontrar as atividades que deverão ser tratadas no contexto do sistema. Ao final da primeira etapa é gerada uma lista de processos e atividades dentro do contexto da solicitação.

#### **2.6.1.2 Etapa 2 – Identificar necessidades dos processos**

A segunda etapa consiste em “especificar as necessidades existentes em cada atividade dos processos no contexto da solicitação” (KNIGHT, 2004. p.70).

O primeiro passo consiste na identificação dos principais problemas que ocorrem na execução das tarefas. Para isso devem ser realizadas entrevistas com os executantes das mesmas. No segundo passo, devem ser identificadas as causas dos problemas identificados. No terceiro passo analisa-se o processo é eficiente e executado corretamente. Neste passo é possível detectar os pontos de melhoria do processo e problemas que estão gerando impacto em outros processos ou nos clientes. Também podem ser identificadas outras características desejáveis a serem adicionadas ao processo como segurança e velocidade. Essas necessidades deverão ser relacionadas com os processos que as originaram.

Ao final dessa etapa é gerada a lista das necessidades existentes em cada atividade do processo.

### **2.6.1.3 Etapa 3 – Identificar os impactos das necessidades**

“O objetivo dessa etapa é obter uma visão dos impactos das necessidades identificadas” (KNIGHT, 2004. p.74). No primeiro passo deve-se identificar as consequências geradas por cada necessidade. Após a identificação, as consequências devem ser analisadas para identificar os objetivos de negócio afetados, no segundo passo. No terceiro passo as pessoas envolvidas no negócio que sofrem as consequências das necessidades existentes, são identificadas e relacionadas com os objetivos do passo anterior. No último passo, as necessidades que serão atendidas devem ser identificadas e priorizadas.

Uma tabela que mostra as necessidades existentes relacionadas com suas consequências, objetivos, pessoas afetadas e prioridades é gerada ao final dessa etapa.

### **2.6.2 Fase 2 - Visão geral da solução**

Com um melhor entendimento do negócio e com as principais necessidades identificadas, a segunda fase do método tem como objetivo “criar a visão da solução que atenda às reais necessidades da solicitação feita, além de deixar claro, para os envolvidos, quais as principais funcionalidades e restrições do sistema” (KNIGHT, 2004. p.77). É possível que o sistema gere mudanças no processo de negócio da empresa. Por isso, ao final da segunda fase deve-se verificar os impactos gerados e atualizar o modelo de negócio caso necessário para que ele continue refletindo a realidade do negócio. Esta fase é composta pela quarta, quinta, sexta e sétima etapa do método.

Na quarta etapa, o objetivo é analisar as necessidades encontradas e definir como elas serão solucionadas. A quinta etapa dedica-se a identificar as fronteiras do sistema. Na sexta etapa “são identificados os impactos causados pelo novo sistema no negócio da organização” (KNIGHT, 2004. p.78). É na sétima e última etapa do método que o documento de requisito de *software* é gerado de acordo com as informações elicitadas.

O resultado gerado ao final desta fase é um resumo das principais características do *software*, que posteriormente deverão ser refinados para gerar os

requisitos do sistema de fato. Ao mesmo tempo, no final da fase, a organização será capaz de perceber as mudanças que ocorrerão em seus processos com a criação do sistema” (KNIGHT, 2004. p.78).

#### **2.6.2.1 Etapa 4 – Identificar as funcionalidades e restrições**

O primeiro passo desta etapa consiste na identificação das funcionalidades e restrições do sistema. Para isso deve-se avaliar o modelo de atividade e o modelo de localização, caso a organização seja distribuída geograficamente, afim de identificar restrições ou funções específicas que o sistema deve prover. “Este passo pode ser executado pelos próprios engenheiros de *software*, com a condição de que ao seu término, as pessoas da organização validem o que foi feito” (KNIGHT, 2004. p.78). No segundo passo, cada necessidade é analisada, considerando o que deve ser acrescentado no sistema para que ela seja suprida.

Ao final da quarta etapa é gerada a lista de todas as funcionalidades e restrições identificadas para o sistema, que possibilita a relação entre as atividades e processos com as restrições e funcionalidades do sistema.

#### **2.6.2.2 Etapa 5 – Identificar as fronteiras do sistema**

De acordo com Knight (2004) o objetivo dessa etapa é identificar os pontos em que o sistema interage com outros recursos do negócio. Deve-se identificar os usuários, meios de armazenamento e outros sistemas que se relacionarão com o sistema.

O primeiro passo tem como objetivo encontrar os papéis que executam as atividades, identificar quais tarefas serão totalmente executadas pelo sistema e quais serão somente apoiadas por ele. O executor do papel deverá acessar as tarefas que forem apenas apoiadas pelo sistema. Atores externos também devem ser considerados, pois podem influenciar o processo e o resultado das atividades. Pode ser necessário adicionar alguns desses atores externos como usuários do sistema afim de gerar as informações necessárias.

No segundo passo avalia-se a necessidade de interação com outros sistemas, verificando se a tarefa continuará a ser executada ou apoiada por esses sistemas externos após a implantação do sistema em desenvolvimento. Caso essa interação ocorra, deve-se definir quais as informações relevantes que deverão ser tratadas nos sistemas.

O terceiro passo é dedicado a analisar o meio de armazenamento. Knight (2004) explica:

“Se o meio de armazenamento acessado por uma atividade for computacional, um banco de dados existente na organização, por exemplo, deverá haver interação deste com o sistema. E, se for um meio de armazenamento manual, e suas informações forem relevantes ao sistema, então o trabalho de desenvolver o sistema englobará o trabalho de criar esse meio de armazenamento também, que evidentemente terá interação com o sistema”

Ao final desta etapa, é gerada uma lista de recursos que interagirá com o sistema, e suas especificações, bem como um diagrama que represente o contexto de utilização do sistema.

### **2.6.2.3 Etapa 6 - Identificar os impactos do novo sistema**

O objetivo desta etapa é identificar como o sistema afeta o negócio. Com a utilização do sistema, podem ocorrer alterações no processo como, por exemplo, algumas tarefas podem passar a ser executadas automaticamente. Isso deve ser identificado e, posteriormente, o modelo de atividades deve ser atualizado.

No primeiro passo identifica-se quais as atividades que serão realocadas para novos executantes, fazendo com que estas não sejam mais responsabilidade de seus antigos donos. No segundo passo, deve-se listar os meios de armazenamento e os papéis que irão interagir com o sistema. Os passos listados, devem ser realizados com a ajuda dos gestores.

Como resultado dessa etapa é gerada uma lista, que representa o conjunto de mudanças que ocorreram no modelo de processos, representando as alterações causadas ao processo com o surgimento do novo sistema.

#### **2.6.2.4 Etapa 7 - Gerar Documento de Requisitos de Software**

A última etapa do método consiste na documentação das informações relevantes ao sistema no Documento de Requisitos de *Software* com base nos resultados gerados nas etapas anteriores.

### **2.7 Scrum**

O Scrum é uma metodologia ágil para gestão e planejamento de projetos de *software*. “É uma metodologia de adaptação, interatividade, rapidez, flexibilidade e eficiência, projetada para favorecer um valor significativo de forma rápida durante todo o projeto” (SCRUMSTUDY, 2016. p. 2). De acordo com a SCRUMSTUDY, os times Scrum são auto gerenciáveis, multifuncionais e trabalham em conjunto durante a Sprint para gerar o entregável. O Time *Scrum* é composto por: (1) *Scrum Master*, que é um facilitador, responsável por ensinar as práticas Scrum e remover impedimentos que afetem a equipe ou o projeto. (2) P.O. -*Product Owner* (dono do produto) – representa a voz do cliente. É ele quem identifica e define as prioridades do projeto de acordo com as necessidades do cliente. (3) Time, que é um grupo multidisciplinar e auto gerenciável responsável por entender os requisitos e por desenvolver os entregáveis.

No início do projeto deve ser gerada a lista dos requisitos macro do sistema, chamada de Backlog. Nele os itens de maior prioridade e melhor detalhados são listados no topo da lista e os itens indefinidos ou de baixa prioridade são listados no final. O desenvolvimento do projeto é realizado em ciclos, chamados de Sprint, que podem durar de 1 a 4 semanas. Antes de iniciar cada Sprint deve ser realizada a reunião de planejamento, na qual o time, P.O. e Scrum Master devem discutir os itens prioritários do Backlog e mensurar o que conseguem entregar no prazo da Sprint. Os itens selecionados do Backlog do projeto, gerarão a lista de backlog da Sprint.

Todas as Sprints do projeto devem ter a mesma duração, entre 1 a 4 semanas, na qual o Scrum Master guia, facilita e protege o time Scrum de impedimentos, afim de criar os entregáveis em um curto espaço de tempo, o que facilita o

acompanhamento do projeto e a validação constante do sistema que está sendo desenvolvido.

Ao final da Sprint é gerado o incremento do produto, ou seja, uma nova parte do sistema que deverá ser incorporada à versão anterior. A Figura 8 ilustra o funcionamento do Scrum.

Figura 7 - Scrum



Fonte: SCRUM ALLIANCE, s.d.

Segundo a SCRUMSTUDY (2016), o Scrum se apoia em três pilares fundamentais: Transparência, Inspeção e Adaptação, sendo:

**Transparência:** Permite que todos os ângulos do projeto possam ser observados por qualquer pessoa. Isso cria uma cultura de trabalho aberta e facilita a colaboração.

**Inspeção:** Os artefatos e o progresso do projeto devem ser inspecionados frequentemente afim de detectar desvios indesejáveis.

**Adaptação:** O projeto deve acompanhar as mudanças de prioridade e necessidades com o objetivo de gerar um produto de *software* que satisfaça à demanda real do cliente.

## 2.8 OpenUP

O OpenUP é um Processo Unificado que aplica a abordagem iterativa e incremental dentro de um ciclo de vida estruturado. Foi criado pela IBM e reflete as boas práticas contidas no RUP, um processo de engenharia de software criado pela *Rational Software Corporation*<sup>9</sup> que tinha o objetivo de garantir a produção de software com qualidade e que atenda às necessidades do cliente, bem como às restrições de escopo e tempo acordados. O OpenUP adota uma filosofia ágil que se concentra na natureza colaborativa do desenvolvimento de *software* e pode ser estendido para abordar uma ampla variedade de tipos de projetos.

Segundo ECLIPSE (2012), o OpenUP é caracterizado por quatro princípios básicos:

- a) Colaborar para alinhar interesses e compartilhar entendimento.
- b) Balancear as prioridades (necessidades e custos técnicos) para maximizar o valor dos interessados.
- c) Focar na articulação da arquitetura para facilitar a colaboração técnica, reduzir o risco, e minimizar o sucateamento e o retrabalho.
- d) Evoluir continuamente para reduzir riscos, demonstrar resultados, e ganhar feedback do cliente.

O processo do OpenUP é dividido em três partes: Ciclo de vida do projeto, ciclo de vida da interação e micro incremento.

O ciclo de vida do projeto é composto de quatro fases:

- a) Fase Concepção: na qual espera-se que todos estejam de acordo com o escopo e objetivos do projeto.
- b) Fase de Elaboração: Nesta fase os riscos devem ser tratados, todos devem estar de acordo com a arquitetura definida e com a entrega de valor para o cliente.
- c) Fase de Construção: Esta fase é dedicada ao desenvolvimento do sistema.

---

<sup>9</sup> Empresa do segmento de software adquirida pela IBM em 2002.

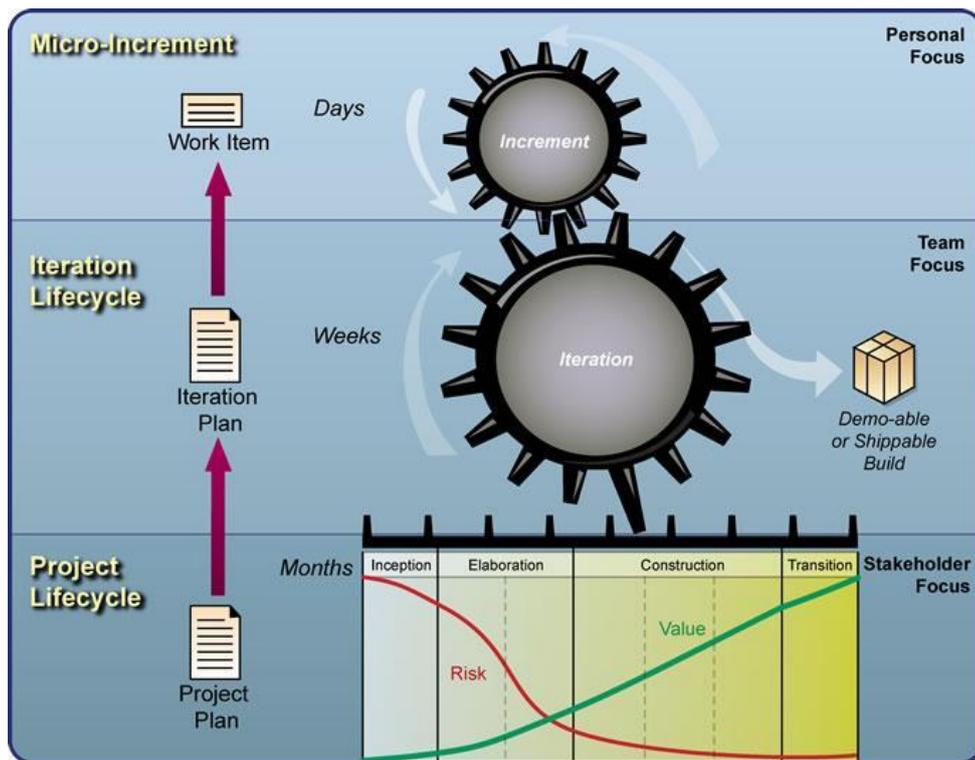
- d) Fase de Transição: Na qual são realizados testes e validações para que o sistema possa ser colocado em produção.

Além da divisão por fases o projeto também é dividido em iterações (chamadas de Sprints na metodologia Scrum), que consistem em ciclos de algumas semanas e ao final de cada ciclo um novo incremento do produto deve ser gerado.

O projeto pode ser acompanhado de maneira mais detalhada a partir do micro incremento, que é a execução de pequenos passos para alcançar o objetivo da iteração. Pode se referir ao produto de trabalho de alguns dias ou horas, produzido por uma pessoa ou grupo.

A Figura 9 demonstra o funcionamento do OpenUP:

Figura 8 - OpenUP

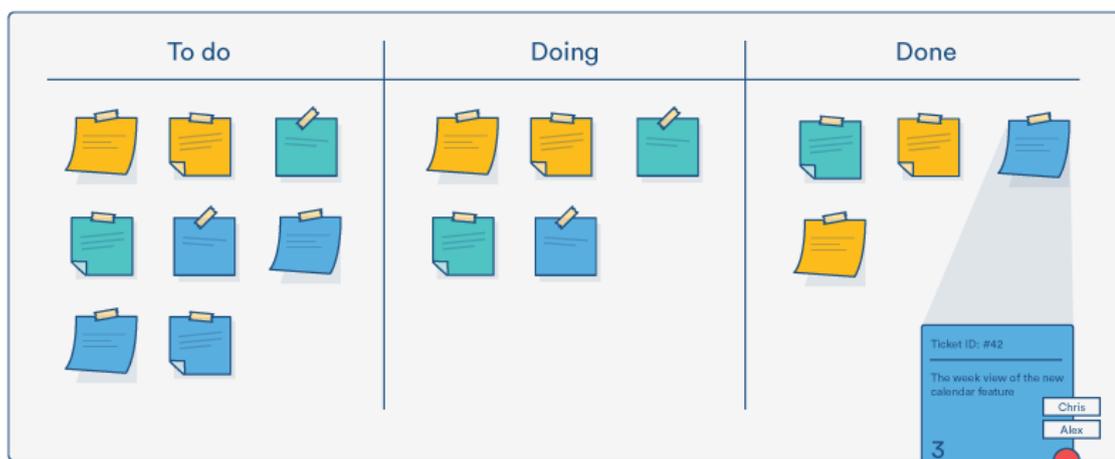


Fonte: [epf.eclipse.org/wikis/openup](http://epf.eclipse.org/wikis/openup) (2012)

## 2.9 Kanban

O Kanban é uma estrutura popular usada por equipes de *software* que praticam o desenvolvimento ágil de *software*. Com ele é possível visualizar o progresso das tarefas e acompanhar o andamento do projeto. Pode ser um quadro físico com cartões, ou um sistema que simule o Kanban. Nele são definidos os status das tarefas e elas devem seguir um fluxo até que sejam concluídas, como ilustrado na Figura 10:

Figura 19 – Kanban



Fonte: <https://br.atlassian.com/agile/kanban>, s.d.

## 2.10 Fábrica de Tecnologias Turing

A FTT, é uma unidade do curso de Engenharia de Computação da Unievangélica criada no ano de 2009. É também um ambiente no qual os alunos do curso podem vivenciar a experiência de trabalhar em projetos reais, com o acompanhamento dos professores, coordenação e diretoria do curso, através dos orientadores de projeto e coordenação da Fábrica. Utiliza uma metodologia híbrida para o processo de desenvolvimento de *software* composta por Scrum e Open Up, conforme apresentado na Figura 11 - Processo da Fábrica de Tecnologias Turing, em anexo. Utiliza também o quadro Kanban como uma ferramenta de organização de atividades para tornar o processo mais transparente e visível.

A FTT propicia um ambiente aberto a discussões e suporte para testar novos métodos e técnicas de desenvolvimento de *software*, tanto por parte dos alunos

envolvidos nos projetos, quanto para alunos que desenvolvem pesquisas. A participação dos professores orientadores e o acompanhamento da coordenação do curso facilitam a troca de experiências e proporciona um melhor aprendizado prático sobre o desenvolvimento de sistemas.

É comum que os alunos que ingressam na FTT após poucos meses sejam selecionados nas empresas da região, comprovando assim que o conhecimento adquirido os diferencia dos demais candidatos às vagas e melhora sua adaptação na empresa. A partir das experiências vividas o aluno é capaz de sugerir soluções para os problemas cotidianos das empresas de desenvolvimento de *software* de maneira mais natural e segura.

Para desenvolver o profissional, a FTT conta com 3 núcleos: 1) O núcleo de seleção, responsável pelos processos seletivos de novos membros, definição do perfil esperado e desenvolvimento de avaliações; 2) O núcleo de capacitação, que trabalha em conjunto com o núcleo de seleção durante os processos seletivos, realizando treinamentos afim de gerar um conhecimento prévio das práticas em fabricas de *software*. 3) O núcleo de desenvolvimento, que atua após a seleção e contratação do candidato, durante o tempo que este participa dos projetos. O núcleo de desenvolvimento tem o objetivo de direcionar os estudos dos membros da FTT, promover troca de experiências e incentivar o crescimento profissional através do conhecimento.

A FTT foi estruturada primeiramente para desenvolver múltiplos projetos com pequenas equipes multidisciplinares para cada um deles, e posteriormente passou por uma reestruturação para o desenvolvimento do *Virtoo*, na qual as equipes passaram a ser divididas de acordo com a função: Requisitos, teste e desenvolvimento.

Até a data desta pesquisa, a FTT já havia entregue 12 projetos de *software* para diversas áreas e estava desenvolvendo um projeto internacional, chamado *Virtoo*, para uma instituição de ensino de Angola.

### **2.10.1 Projeto Virtoo**

O projeto Virtoo foi desenvolvido com o objetivo de atender às necessidades de alunos e funcionários dos institutos ISPVIDA e ISTEEL da Angola que solicitaram um sistema capaz de gerenciar as ações realizadas nos institutos.

De acordo com a declaração de visão do projeto o software consistia em um sistema web para gestão acadêmica e financeira que possibilitava que alunos e funcionários das instituições realizassem suas atividades acadêmicas, com interface intuitiva e responsiva, possibilitando assim o acesso através de dispositivos móveis. O objetivo do produto era agilizar e simplificar o processo de gestão acadêmica e financeiro apresentando uma qualidade satisfatória na apresentação das informações referentes às instituições.

### **3 RESULTADOS**

#### **3.1 Necessidades na elicitação e documentação dos requisitos**

Durante o desenvolvimento do projeto Virtoo – sistema de gestão acadêmica e financeira para uma universidade de Angola, percebeu-se que a documentação tradicional apesar de utilizar um formato já adequado ao contexto da FTT, ainda apresentava complicações em apresentar as necessidades a serem atendidas pelo sistema de forma clara e para acompanhar as mudanças do projeto devido ao número de documentos que apresentavam as informações de um mesmo caso de uso. Outra dificuldade encontrada foi com relação ao baixo entendimento do projeto e seu contexto de aplicação, por se tratar de um projeto internacional e o contato com o cliente ser reduzido, além da pouca experiência da equipe do projeto.

Para aplicação e avaliação do modelo foram realizadas quatro etapas, sendo: 1) adaptação de um caso de uso ao modelo proposto, utilizando BPMN; 2) avaliação da adaptação realizada; 3) transcrição dos demais documentos para o novo formato; 4) desenvolvimento dos requisitos documentados com BPMN, avaliação dos resultados e identificação de melhorias.

#### **3.2 Etapas para a aplicação e avaliação de documentação com BPMN**

A primeira etapa foi adaptar a documentação que já havia sido desenvolvida ao novo modelo de documentação. No primeiro passo definiu-se a ferramenta para a modelagem do processo e desenvolvimento da documentação. A ferramenta selecionada foi o Bizagi, pois ela possibilita a navegação pelo fluxo e a descrição de todos os tópicos necessários para o entendimento dos requisitos do sistema, além de gerar a documentação em formato HTML, possibilitando a navegação pelas atividades e visualização da documentação relacionada. O segundo passo foi revisar e definir as informações necessárias para documentar os requisitos e configurar a ferramenta para criar as atividades com os tópicos definidos. No passo três, foi selecionado um caso de uso já documentado para ser adequado à nova modelagem. Este foi transcrito para o modelo com BPMN e apresentado com o modelo navegável.

A segunda etapa consistiu na avaliação da adaptação realizada afim de definir o formato do modelo a ser seguido. Como primeira atividade dessa etapa, o exemplo da nova documentação foi apresentado e avaliado pelos analistas de requisito. Após a aprovação por parte deles, uma nova reunião com toda a equipe, foi realizada. Nesta reunião os participantes apresentaram as dificuldades que tinham com o modelo antigo, como: dificuldade para encontrar todas as informações de um caso de uso, a divisão das informações do caso de uso em vários documentos, dificuldade em solicitar e receber ajustes necessários da documentação, como por exemplo atualização de recursos de tela apresentados nos protótipos devido a limitações técnicas. A partir das observações feitas pela equipe o exemplo foi ajustado e solução proposta: um modelo de documentação utilizando BPMN, no qual todas as informações do caso de uso estariam contempladas na documentação de uma das tarefas do fluxo do processo do sistema. Houve também um treinamento sobre a utilização da nova documentação, incluindo a apresentação sobre a ferramenta de modelagem, o Bizagi, a forma de documentação das tarefas e exportação do processo para arquivos web.

Na terceira etapa, os demais documentos casos de uso já existentes foram transcritos para o novo formato, utilizando processo de negócio. E na etapa quatro foi realizado o desenvolvimento da documentação dos novos requisitos do projeto no formato utilizando processo de negócio, a avaliação dos resultados da aplicação do modelo e a definição de melhorias.

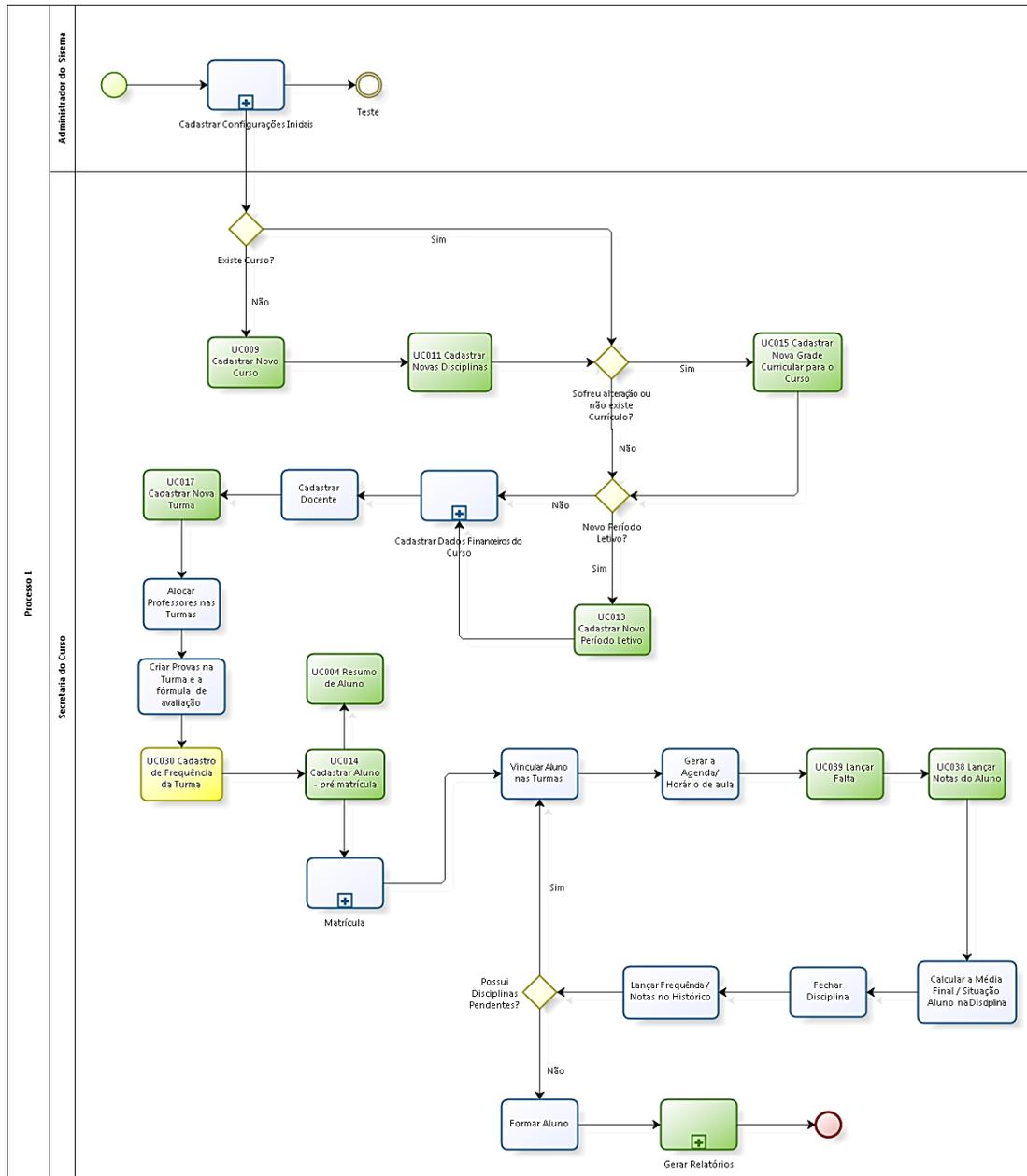
### **3.3 Proposta de nova forma de documentação**

A documentação proposta foi modelada com o auxílio da ferramenta de modelagem de processos Bizagi. Foi realizada a diagramação do processo macro do sistema, apresentando as etapas desde configurações iniciais até a geração de relatórios como demonstra a Figura 12.

Cada tarefa do *Business Process Model and Notation* (BPMN) representa um caso de uso ou um passo no processo do sistema. Primeiramente foi modelado o fluxo, contemplando todas as funcionalidades previstas até a geração de relatórios. Depois os casos de uso foram documentados a partir dos recursos de documentação de tarefa disponíveis na ferramenta. Cada caso de uso já documentado foi marcado de verde. Os demais casos de uso foram apresentados

no diagrama, mesmo que não documentados, para apresentar uma visão geral do processo do sistema.

Figura 10 - Processo do sistema acadêmico Virtoo



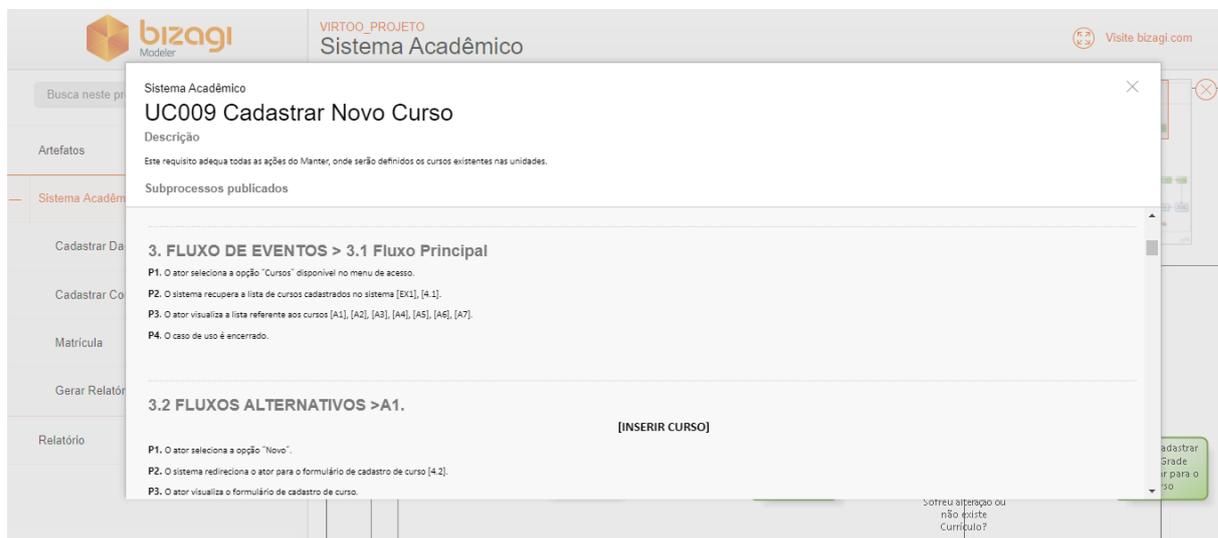
Fonte: Documentação do projeto Virtoo (2017)

A ferramenta de modelagem possibilita a geração de uma página HTML para visualização e navegação pelo processo. Com isso a documentação se tornou mais dinâmica e interativa. O processo navegável pode ser acessado em: (<http://virtoo.unievangelica.edu.br/docs>).

Ao selecionar a opção de visualização de detalhes da tarefa, a página exibe a documentação do caso de uso, que segue o padrão de descrição aplicado no modelo padrão conforme a Figura 13, composto por:

- Histórico de Revisão
- Atores
- Pré-Condições
- Fluxo de Eventos
- Fluxo Principal
- Fluxos Alternativo.
- Fluxos de Exceção
- Protótipos
- Definição de Campos
- Descrição de Comandos
- Regras de Apresentação

Figura 11 – Modal de detalhes da tarefa



Fonte: Processo do projeto Virtoo

### 3.4 Comparação dos modelos

Para identificar e comparar os resultados obtidos a partir da utilização de cada modelo foram realizadas observações *in loco*, entrevistas e aplicação do questionário (Apêndice A). Os dados e informações coletados foram analisados e a

partir dos pontos de melhoria apresentados no item 3.5, o modelo de documentação foi ajustado e proposto como objeto de pesquisas futuras.

### **3.4.1 Questionário e entrevistas**

O questionário foi aplicado aos integrantes que participaram do projeto durante a aplicação da documentação padrão, que participaram da transição para o modelo baseado em processo de negócio e trabalharam pelo menos 1 *sprint* com o novo modelo. Para a aplicação do questionário foram selecionados membros das três equipes: requisitos, desenvolvimento e testes, e que participaram do projeto tanto durante a utilização da documentação com caso de uso quanto durante a documentação com BPMN, totalizando 30% dos integrantes do projeto.

As questões foram criadas de modo que avaliassem os itens para uma boa documentação de requisitos de *software*. De acordo com a IEE-830 um documento de especificação de *software* deve ser: “Correto; Não ambíguo; Completo; Consistente; Classificável por importância e/ou estabilidade; Verificável; Modificável; Rastreável;” (GONÇALVES, 2004, p. 6).

Foram realizadas 2 questões para comparar a documentação com caso de uso e a documentação com BPMN. A primeira delas, avaliando 10 itens relacionados às recomendações da IEEE, na qual os entrevistados deveriam escolher qual dos modelos melhor se adequava à afirmação. Já a segunda questão questionava quais os problemas enfrentados na elicitação de requisitos de acordo com a opinião do entrevistado.

Adicionalmente, entrevistas foram realizadas com o objetivo de identificar pontos subjetivos da utilização do modelo e também observar melhor os desafios e melhorias apresentados durante o processo de adaptação e aplicação do novo modelo. Os entrevistados foram questionados sobre a percepção da aplicação do novo modelo e à utilização do modelo anterior, os benefícios, desafios e sugestões de melhoria. Diferentemente do questionário que seguia um roteiro específico as entrevistas foram realizadas de forma mais livre para evitar induzir as respostas.

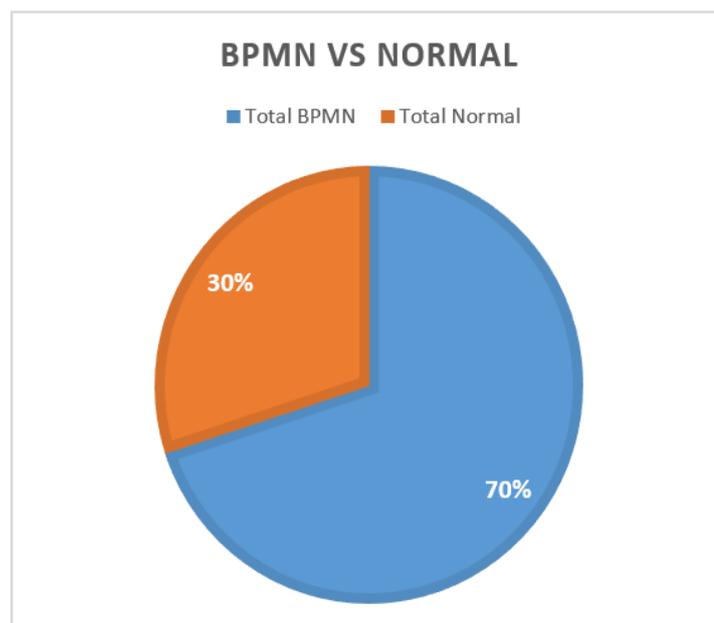
### 3.4.2 Dados obtidos

As entrevistas e observações apontam que a maior parte dos entrevistados considera que a documentação com BPMN está mais aderente aos aspectos apresentados pela IEEE-830, sendo que parte dos entrevistados chegou a citar que com a utilização do novo modelo ficou mais fácil de entender o objetivo do caso de uso, suas dependências e o comportamento do sistema, devido a visualização do fluxo do sistema.

No questionário, os entrevistados afirmam que a principal dificuldade no levantamento de requisitos está relacionada às mudanças nos requisitos, e a segunda maior dificuldade foi apontada referente tanto à dificuldade em validar o que o cliente solicitou quanto à dificuldade do cliente expressar o que realmente deseja. Isso ocorre pelo fato de se tratar de um projeto desenvolvido à distância, à complexidade do projeto e também a fatores relacionados à experiência da equipe.

Segundo os dados coletados, 70% das respostas apontam a documentação BPMN como sendo mais aderente às regras da IEEE quando comparada com a documentação padrão. Como mostra o gráfico 01.

Gráfico 01 - Modelos e sua aderência às recomendações da IEEE

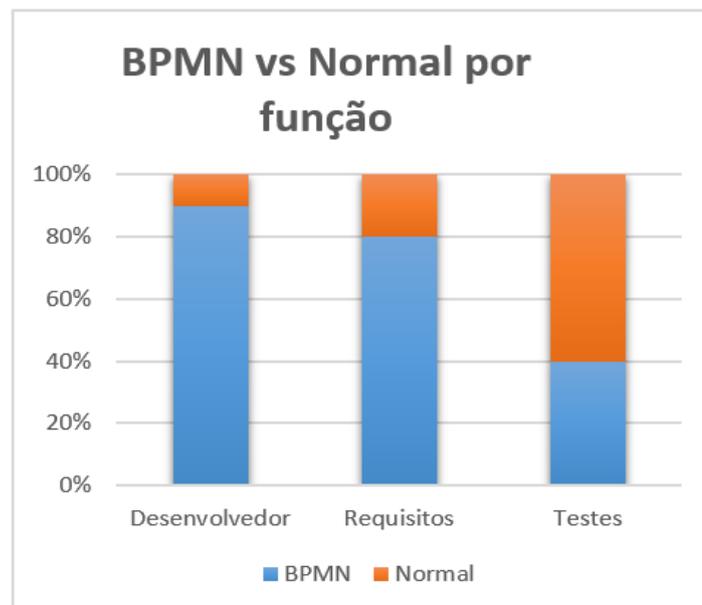


Fonte: Autores, 2017

A maior parte dos analistas de requisitos e desenvolvedores responderam que a documentação com BPMN atende à maioria dos requisitos para uma boa documentação. Já entre os analistas de teste a opinião ficou dividida, sendo que em somente 40 % dos casos a documentação com BPMN foi apontada como melhor, como mostra o gráfico 02.

Parte das avaliações negativas são devido à maior dificuldade para transcrever o modelo em BPMN para os casos de teste, visto que os casos de uso descritos da forma anterior facilitavam a criação dos casos de teste por já seguirem uma estrutura semelhante ao documento de caso de teste que deveria ser gerado.

Gráfico 02 - Comparativo de respostas pela função do entrevistado

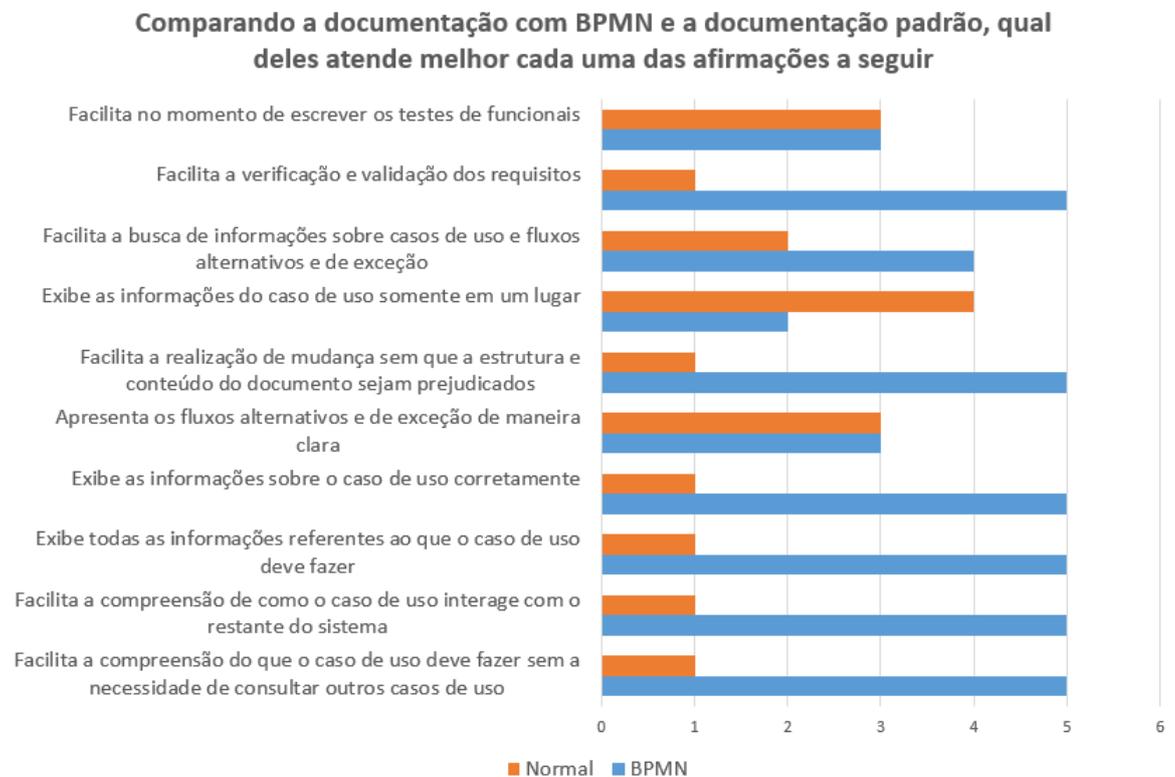


Fonte: Autores, 2017

De acordo com o Gráfico 03, a documentação com caso de uso apresenta as informações de forma mais centralizada. Nos quesitos “apresentam fluxos alternativos e de exceção de maneira mais clara” e “Facilita no momento de escrever os testes funcionais” a documentação com caso de uso e com BPMN foram votadas igualmente, apontando que nesses dois pontos não houve melhoria com relação à documentação com caso de uso. Isso se deu devido ao fato de que os fluxos de detalhamento dos casos de uso ainda seguem a estrutura anterior, o que pode estar diminuindo o desempenho da aplicação da documentação já que com a modelagem

padrão, segundo os entrevistados, a descrição dos fluxos apresentava passos óbvios e em alguns casos deixava de lado pontos importantes do fluxo do caso de uso. Em ambos os pontos com baixa avaliação, a inexperiência da equipe também influencia, comprometendo o total entendimento e descrição dos fluxos alternativos e de exceção e afetando na criação de casos de testes.

Gráfico 03 - Comparando a documentação com BPMN e a documentação padrão com relação às características recomendadas pela IEEE



Fonte: Autores, 2017

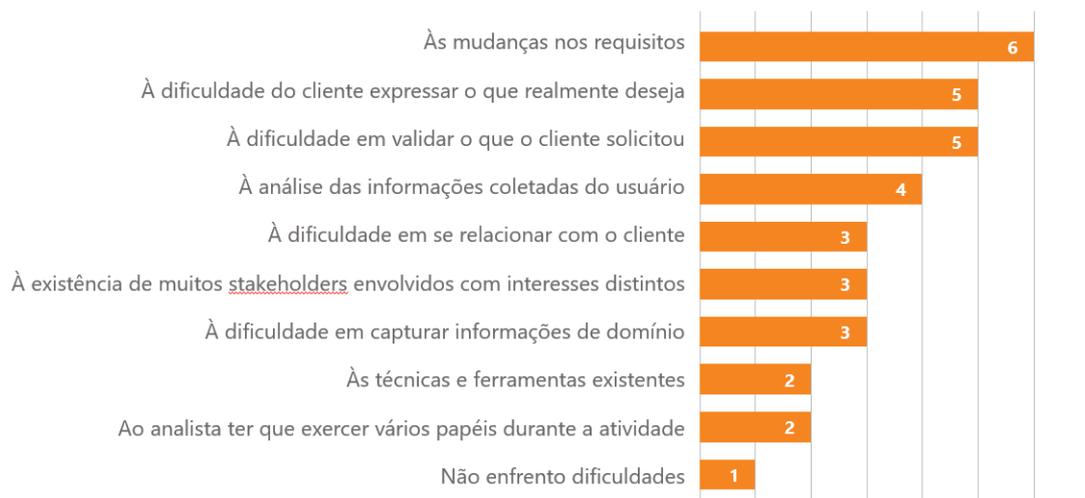
A segunda pergunta do questionário avaliou os principais problemas no levantamento de requisitos. Todos os entrevistados apresentaram a mudança dos requisitos como a principal dificuldade na elicitação, que acontece tanto devido à complexidade do projeto quanto devido à inexperiência dos analistas que tem dificuldade em levantar as necessidades do caso de uso de modo completo no

primeiro momento, sendo necessários ajustes posteriores para incorporar as necessidades encontradas ao longo do tempo.

Os entrevistados apontaram também a “dificuldade do cliente expressar o que realmente deseja” e a “dificuldade em validar o que o cliente solicitou” como outro fator prejudicial, como mostra o gráfico 04. Isso reforça a necessidade de um modelo de documentação de requisitos que proporcione uma apresentação clara do que o sistema se propõe a realizar, bem como facilite no processo de adequação e atualização do documento, como é a proposta da documentação com BPMN.

Gráfico 04 - Principais problemas no levantamento de requisitos

### Principais problemas no levantamento de requisitos



Fonte: Autores, 2017

Os analistas de testes foram os que mais informaram fatores prejudiciais para levantamento de requisitos, representando 47% de todos os problemas reportados, já os próprios analistas de requisitos informaram um número menor de problemas enfrentados, apenas 29%. Ainda assim os desenvolvedores foram os que menos relataram problemas, num total de 24% dos problemas reportados. O gráfico 05 apresenta o total de problemas reportados de acordo com o papel do entrevistado.

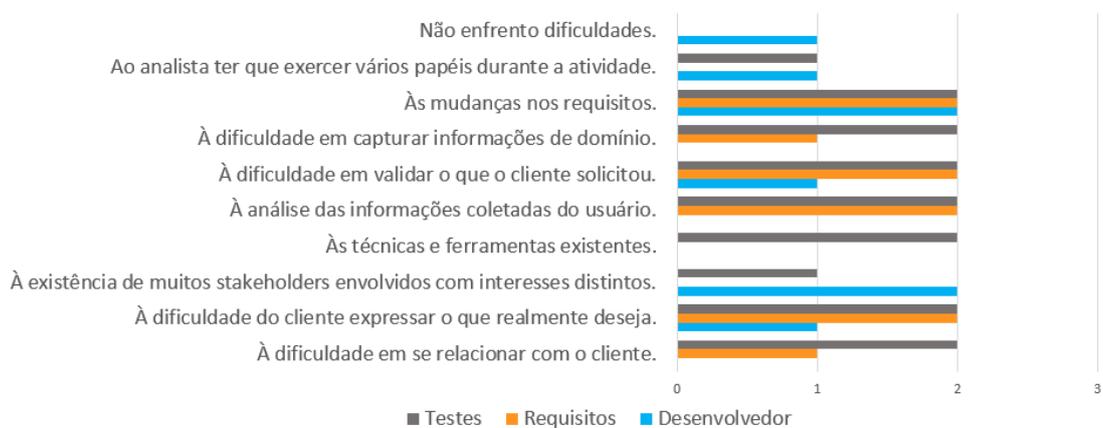
Gráfico 05 - Total de problemas reportados de acordo com a função do entrevistado



Fonte: Autores, 2017

De acordo com os dados apresentados no gráfico 06, os analistas de testes foram os únicos que marcaram “As técnicas e ferramentas existentes” como um problema no levantamento de requisitos, ao passo que os analistas de requisitos não consideram este um fator que afeta o levantamento de requisitos.

Gráfico 06 - Problemas reportados, de acordo com a função do entrevistado



Fonte: Autores, 2017

Foi relatado durante as entrevistas que apesar da documentação com BPMN ser mais intuitiva, o tempo para a criação continuou extenso. Isso se deu, em partes, devido à pouca experiência de boa parte da equipe de analistas de requisito, devido à mudança dos membros do time do projeto.

Outro fator que impactou negativamente na adaptação do novo modelo foi a falta de conhecimento de processos por parte dos analistas de requisitos, reforçando o que foi afirmado por Machado: que ainda existe um déficit no ensino sobre processos de negócio, e isso faz com que os analistas saibam como criar casos de uso e outros tipos de documentos de especificação de requisitos mas não consigam alinhar as demandas do sistema com os processos da empresa e não entendem o que ele significa no contexto do negócio do cliente (MACHADO, 2011).

### **3.5 Pontos de melhoria identificados**

Após a análise das informações coletadas foram identificados alguns pontos de melhoria. Dentre eles está a necessidade de treinamento da equipe de analistas sobre BPMN e a modelagem do fluxo também no detalhamento do fluxo do caso de uso, ao invés de detalhar somente o nível macro, apresentando as interações das funcionalidades.

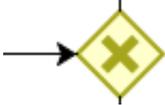
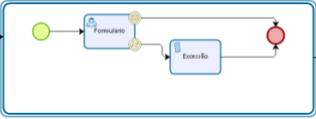
Visando melhorar o modelo, deixá-lo mais aderente e de fácil entendimento foram realizados ajustes e proposto um novo modelo para pesquisas futuras, como apresentado no tópico a seguir.

### **3.6 Proposta de melhoria**

O modelo melhorado segue a mesma estrutura do modelo anterior, porém o detalhamento do caso de uso deixou de ser apenas um modal descritivo, como a figura 16 e passou a ter seu fluxo modelado, conforme a Figura 16.

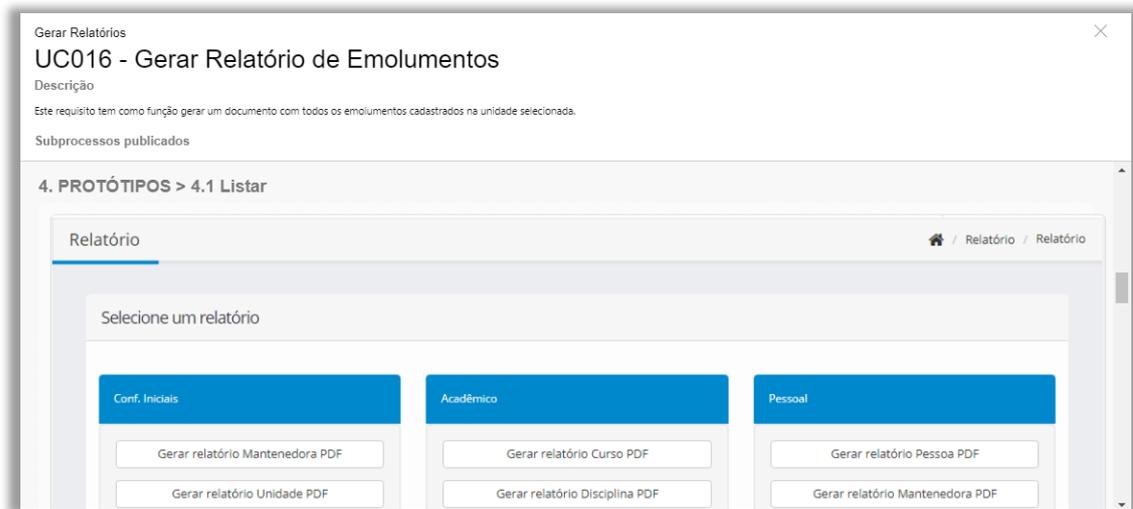
Em cada tarefa do fluxo está a documentação do passo, com suas regras, protótipos de tela e regras de apresentação. A tabela 1, abaixo, apresenta os elementos do BPMN e seu significado no modelo proposto.

Tabela 2 - Elementos BPMN e significado no modelo proposto

Elemento	Definição e aplicação no modelo proposto
	<p>Elemento de início.</p> <p>Assim como no BPMN, representa o início de um processo.</p>
	<p>Tarefa do tipo chamada de serviço:</p> <p>Representa uma tarefa realizada pelo próprio sistema (Virtuo). Na imagem, o sistema é responsável por listar as mantenedoras.</p>
	<p>Gateway exclusivo:</p> <p>Representa um ponto de decisão no fluxo.</p> <p>Ele contém somente uma entrada e pode exibir várias saídas. As saídas representam as opções disponíveis. Somente um dos caminhos deve ser escolhido.</p>
	<p>Tarefa de usuário:</p> <p>Usada para representar tarefas que serão executadas pelo usuário.</p> <p>Nessas tarefas são detalhados os protótipos de tela, detalhamento de campos, botões e regras de apresentação.</p>
	<p>Sub processo:</p> <p>Representar um processo que deve ser realizado dentro de outro processo. Na imagem ao lado, o sub processo 'cadastrar mantenedora' está contido dentro do processo 'Manter mantenedora' que, por sua vez, faz parte do processo macro do sistema.</p>
	<p>Elemento de borda do tipo ERRO:</p> <p>Utilizado para representar que ocorreu um erro no momento da execução da tarefa.</p>

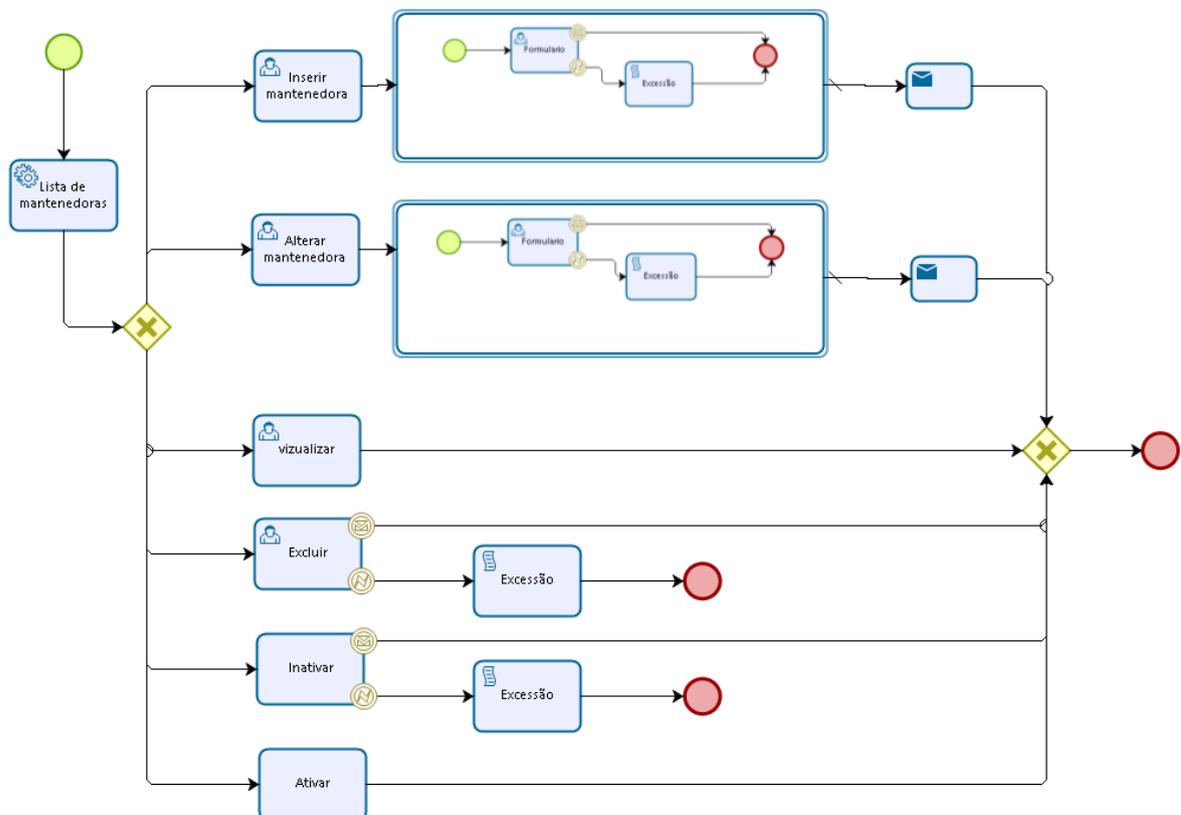
	<p>Tarefa de mensagem:</p> <p>Utilizada no modelo para exibir as mensagens de sucesso após as ações de cadastro e edição.</p>
	<p>Elemento final:</p> <p>Representa o final do processo.</p>

Figura 12 – Modal de detalhes de um caso de uso – modelo anterior



Fonte: Autores,2017

Figura 13 – Caso de uso Manter Mantenedora, modelado de acordo com a proposta



Fonte: Autores, 2017

O processo apresenta os fluxos alternativos e fluxos de exceção modelados. Na imagem acima, no caso de uso Manter Mantenedora, a primeira ação realizada é a listagem de mantenedoras (tarefa realizada pelo sistema e simbolizada pela engrenagem). Logo em seguida o sistema oferece seis opções para o usuário: Inserir, alterar, visualizar, excluir, inativar, ativar. No caso das tarefas 'inserir mantenedora' e 'alterar mantenedora', o sistema apresenta um sub processo, que exibe um formulário a ser preenchido pelo usuário e pode exibir uma mensagem ou um erro, de acordo com a ação do usuário e então finaliza o sub processo. Ao finalizar o sub processo de cadastro o sistema exibe a mensagem de sucesso.

No caso das opções 'Excluir Mantenedora' e 'Visualizar Mantenedora' existem duas opções de saída: ou uma mensagem, caso a ação seja bem-sucedida ou um elemento de erro, caso haja algum fluxo de exceção a ser tratado.

## 4 Conclusão

Durante a realização desta pesquisa foi identificada a documentação de requisitos utilizada pela equipe do projeto Virtoo, desenvolvido pela FTT. A partir dos relatos e das observações realizadas foi definido um novo modelo de documentação baseado na modelagem do processo do sistema e no detalhamento dos casos de uso que o compõem. Após os ajustes para a primeira versão do modelo, ele foi aplicado na FTT durante 6 meses e avaliado a partir de entrevistas e questionários baseados nos critérios para um bom documento de especificação definidos pela IEEE-830<sup>10</sup>.

A partir da aplicação e avaliação comparativa com relação ao modelo anterior percebeu-se que o novo modelo gerou melhoria no entendimento dos requisitos ao longo do ciclo de desenvolvimento e o fato de ter as informações em um documento unificado fez com que a leitura e consulta da documentação ficasse mais simples, já que era necessário consultar quatro documentos simultaneamente para obter o entendimento completo do requisito. Adicionalmente, a apresentação do processo do sistema promoveu um melhor entendimento das dependências dos requisitos e uma melhor visualização do escopo do sistema, melhorando assim a qualidade da documentação dos requisitos e facilitando seu entendimento.

Apesar disso, identificou-se que o detalhamento do processo deve ser demonstrado também no nível do fluxo do caso de uso, para garantir a satisfatoriedade da aplicação do modelo. Adicionalmente é necessário treinar a equipe com relação à notação BPMN e à aplicação de processos. Também foi identificado que na equipe de teste a documentação teve pouca aderência devido à maior necessidade de análise ao criar os casos de teste.

### 4.1 Trabalhos futuros

A partir dos resultados dessa pesquisa foi possível identificar possíveis melhorias no modelo e na sua aplicação, relacionadas a treinamento da equipe e ao nível de detalhamento do processo. Com o objetivo de aperfeiçoá-lo e validar a melhoria proposta, é recomendada a sua aplicação nas próximas sprints do projeto Virtoo.

---

<sup>10</sup> Norma IEE-830, disponível em: <http://www.urisan.tche.br/~pbetencourt/engsoft/IEEE830/index.html>

## REFERÊNCIAS BIBLIOGRÁFICAS

ARAÚJO, Luís César G. de; GARCIA, Adriana Amadeu; MARTINES, Simone. **Gestão de processos: melhores resultados e excelência organizacional**. 2.ed. São Paulo: Atlas, 2011.

BRITTO, Gart Capote de. **Guia para Formação de Analistas de Processos - BPM**. 1.ed. Rio de Janeiro: Gart Capote, 2011.

ECLIPSE. **OpenUp**. (2012). Disponível em: <http://epf.eclipse.org/wikis/openup/>. Acesso em 19 de dez de 2017.

FILHO, Wilson de Pádua Paula. **Engenharia de software: fundamentos, métodos e padrões**. 3. ed. Rio de Janeiro: LTC, 2013.

GHEZZI, Carlo; JAZAYERI, Mehdi, MANDRIOLI Dino. **Fundamentals of Software Engineering. Fundamentals of software engineering**. Prentice Hall PTR, 2002.

GONÇALVES, André; et al. **IEEE Std 830 Prática Recomendada Para Especificações de Exigências de Software**. 2004.

KNIGHT, Débora Mac; ARAÚJO, Renata; BORGES, Marcos. **Elicitação de Requisitos a partir do Modelo de Negócio**. Dissertação de Mestrado, Programa de Pós-Graduação em Informática, NCE/IM-UFRJ, Rio de Janeiro, 2004.

MACHADO, Felipe Nery Rodrigues. **Análise e gestão de requisitos de software: onde nascem os sistemas**. São Paulo: Érica, 2011.

OBJECT MANAGEMENT GROUP. **About OMG®**. Disponível em: <http://www.omg.org/gettingstarted/gettingstartedindex.html>. Acesso em 17 de maio de 2017.

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016.

SCRUMSTUDY. **A Guide to the Scrum Body of Knowledge (SBOK™ Guide) – 2016 edition**. Disponível em: <http://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016.pdf> . Acesso em 24 de maio de 2017.

SOMMERVILLE, Ian. **Engenharia de software**. 8. ed. São Paulo, SP: Pearson Prentice Hall, 2007.

GUIA da Fábrica de Tecnologias Turing. Anápolis, GO. 2016.

ATLASSIAN. **Kanban**. Disponível em: <https://br.atlassian.com/agile/kanban>. Acesso em 18 de dez de 2017.

SCRUM ALLIANCE. **Learn About Scrum**. Disponível em:  
<https://www.scrumalliance.org/why-scrum>. Acesso em 18 de dez de 2017.

GONÇALVES Andre, et al. **IEEE Std 830 Prática Recomendada Para Especificações de Exigências de Software**: Standard Internacional, 2004.  
Disponível em: <http://www.urisan.tche.br/~pbetencourt/engsoftl/IEEE830/> . Acesso em 18 de dez de 2017.

## APÊNDICES

### APÊNDICE A - Questionário

1 - Comparando a documentação com BPMN e a documentação padrão, qual deles atende melhor cada uma das afirmações a seguir: \*

	Documentação o normal	Documentação com BPMN
Facilita a compreensão do que o caso de uso deve fazer sem a necessidade de consultar outros casos de uso		
Facilita a compreensão de como o caso de uso interage com o restante do sistema		
Exibe todas as informações referentes ao que o caso de uso deve fazer		
Exibe as informações sobre o caso de uso corretamente		
Apresenta os fluxos alternativos e de exceção de maneira clara		
Facilita a realização de mudança sem que a estrutura e conteúdo do documento sejam prejudicados		
Exibe as informações do caso de uso somente em um lugar		
Facilita a busca de informações sobre casos de uso e fluxos alternativos e de exceção		
Facilita a verificação e validação dos requisitos		
Facilita no momento de escrever os testes de funcionais		

2 - Na sua opinião as principais dificuldades no levantamento de requisitos estão relacionadas: \*

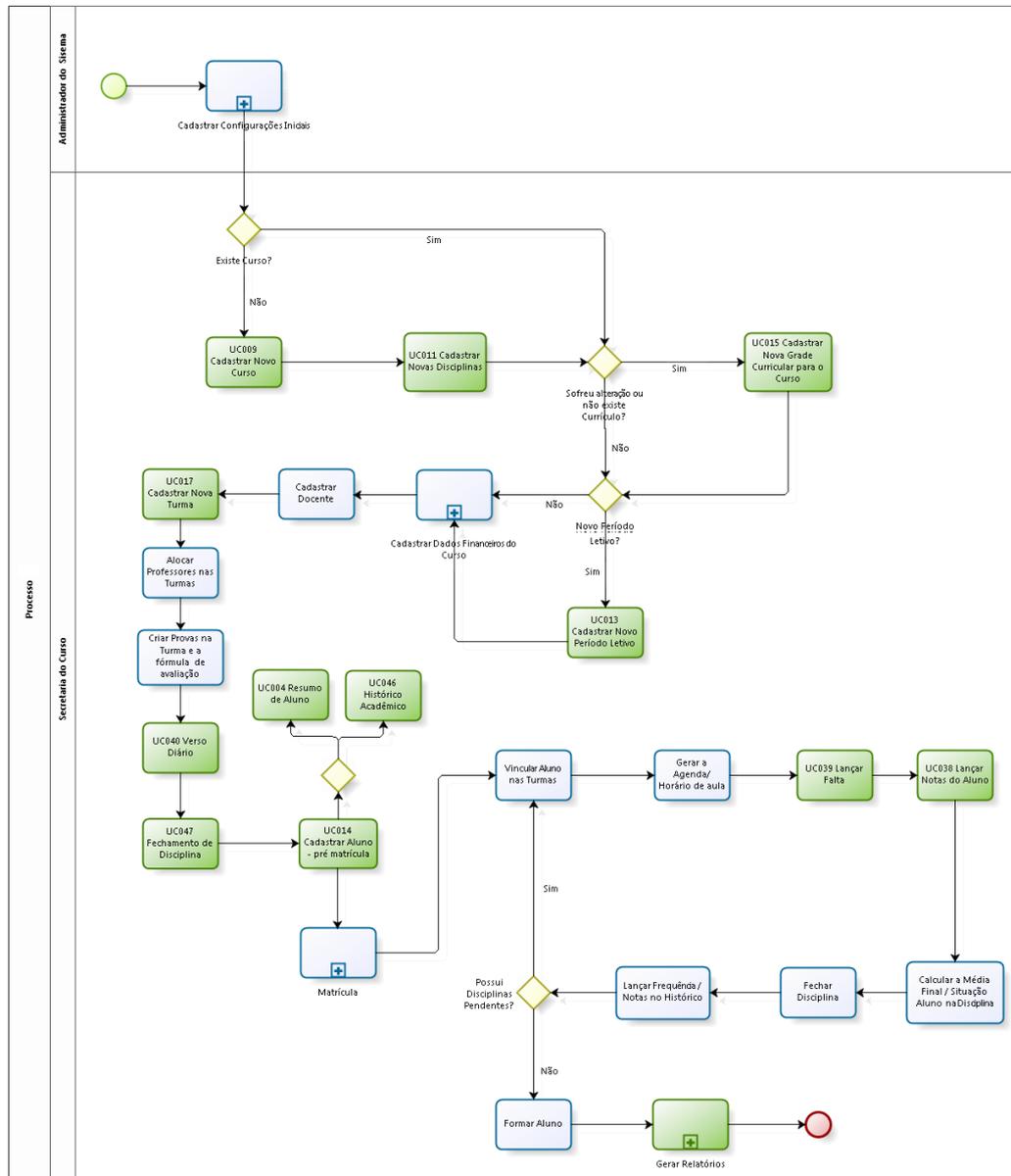
- À dificuldade em se relacionar com o cliente.
- À dificuldade do cliente expressar o que realmente deseja.
- À existência de muitos stakeholders envolvidos com interesses distintos.
- Às técnicas e ferramentas existentes.
- À análise das informações coletadas do usuário.
- À dificuldade em validar o que o cliente solicitou.
- À dificuldade em capturar informações de domínio.
- Às mudanças nos requisitos.
- Ao analista ter que exercer vários papéis durante a atividade.
- Não enfrento dificuldades.
- Outro: \_\_\_\_\_

3 – Observações

## Anexos

## Documento do projeto Virtoo com modelagem de processo

Figura 14 – Fluxo principal do sistema Virtoo

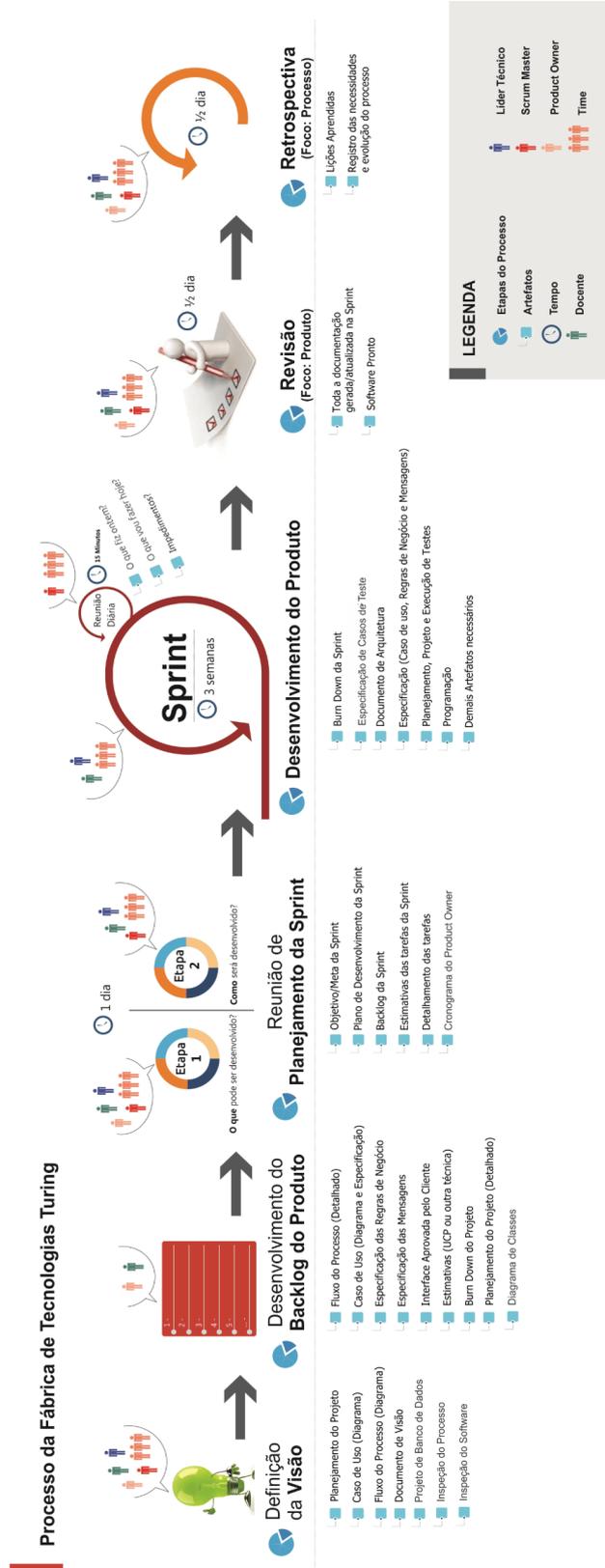


Fonte: GUIA ..., 2015

A documentação pode ser acessada no link: <http://virtoo.unievangelica.edu.br/docs><sup>11</sup><sup>11</sup> Acesse preferencialmente nos navegadores Google Chrome e Firefox

# Processo da Fábrica de Tecnologias Turing

Figura 15 - Processo da Fábrica de Tecnologias Turing



Fonte: GUIA ..., 2015