

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
ESPECIALIZAÇÃO EM QUALIDADE DE SOFTWARE

Lúcio Mosená Soares

ELICITAÇÃO DE REQUISITOS BASEADO EM MÉTODOS ÁGEIS:
UM ESTUDO DE CASO

São Leopoldo

2017

UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
ESPECIALIZAÇÃO EM QUALIDADE DE SOFTWARE

Lúcio Mosená Soares

ELICITAÇÃO DE REQUISITOS BASEADO EM MÉTODOS ÁGEIS:
UM ESTUDO DE CASO

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do título de Especialista em Qualidade de Software, pelo curso de Pós-Graduação Lato Sensu em Qualidade de Software da Universidade do Vale do Rio dos Sinos – UNISINOS.

Orientador: Prof. Dr. Sílvia César Cazella

São Leopoldo

2017

ELICITAÇÃO DE REQUISITOS BASEADO EM MÉTODOS ÁGEIS: UM ESTUDO DE CASO

Lúcio Mosená Soares¹

¹Universidade do Vale do Rio dos Sinos (Unisinos) – São Leopoldo – RS – Brasil

luciomosena@gmail.com

Abstract. *Elicitation of requirements is one of the constant challenges in business processes. The objective of this article is to suggest an improvement in the requirements elicitation stage, using an artifact based on agile methods, focusing on the needs of the organization. A case study was carried out in the information technology department of an industrial sector organization regarding the elicitation of software requirements, highlighting its strengths and weaknesses in this stage of requirements engineering.*

Resumo. *A elicitação dos requisitos é um dos desafios constantes nos processos de empresas. O objetivo deste artigo é sugerir um aprimoramento na etapa de elicitação de requisitos, frente à utilização de um artefato baseado em métodos ágeis, com foco nas necessidades da organização. Um estudo de caso foi realizado no departamento de tecnologia da informação de uma organização do setor industrial, com relação a elicitação dos requisitos de software, evidenciando seus pontos fortes e fracos nesta etapa da engenharia de requisitos.*

Palavras-chave: *levantamento de requisitos, elicitação de requisitos, engenharia de software.*

1. Introdução

No processo de transcrição dos requisitos de usuário e de sistema, grande parte das organizações utilizam um artefato comumente denominado de documento de requisitos. Sommerville (2011) comenta que os requisitos de usuário e de sistema possuem a obrigação de serem evidentes, compreensíveis e, que atendam à todas às qualidades exigidas. No entanto, dificilmente isso acontece na prática, pois as partes interessadas compreendem os requisitos de várias formas, que por sua vez, geram conflitos intrínsecos a estes, necessitando uma nova análise de negócio [SOMMERVILLE 2011]. Como consequência, o prazo de entrega do produto de *software* é afetado e, para que este seja entregue, as fases de verificação dos requisitos, desenvolvimento e testes de sistema terão menos tempo para serem executadas.

Os requisitos de usuário precisam ser descritos de modo que sejam claros e de fácil leitura, tanto para os usuários leigos do sistema, quanto para os que tenham conhecimentos técnicos, ou seja, deve-se evitar o uso da linguagem característica de *software*¹, como artefatos de modelagem de *software*. Deve-se fazer uso da linguagem natural, complementada por tabelas e diagramas, sendo estes simples e intuitivos [SOMMERVILLE 2011]. Já os requisitos de sistema, além da linguagem natural, possuem particularidades, e costumam explicar como os requisitos de usuário serão

¹ Conjunto de instruções que controlam o funcionamento de um computador.

deferidos, como por exemplo, especificações matemáticas, dependências de informações, regras lógicas, entre outros [SOMMERVILLE 2011].

Tanto Kawai (2005), quanto Sommerville (2011), atentam para a variedade de interpretações que a utilização da linguagem natural implica, pois sua conotação remete à ambiguidade das palavras, de cunho flexível, uma vez que algo pode ser dito de maneiras completamente diferentes.

De acordo com Young (2002), calcula-se que um percentual de 85% dos defeitos nos *softwares*, tem sua origem nos requisitos. Para Sommerville (2011), a correção destes defeitos, fazendo uma mudança no sistema após sua implementação, implica em um custo muito maior do que uma correção na fase de elicitação e validação dos requisitos. Isso porque uma mudança de requisitos significa que, tanto o projeto quanto sua codificação irão sofrer mudanças, e o sistema deverá ser testado outra vez [SOMMERVILLE 2011].

Para Sommerville (2011), principalmente a atividade de elicitar requisitos, é focada nas pessoas, e estas, não reagem positivamente com relação à existência de modelos rígidos de sistema, pois sejam quais forem os sistemas, os requisitos mudam. Estas mudanças, por sua vez, intensificam o motivo pelo qual a elicitação dos requisitos é considerada uma das fases mais onerosas e críticas no âmbito da engenharia de requisitos, pois ela é a base para as atividades posteriores, e, além disso, interfere diretamente no produto [SOMMERVILLE 2011].

Considerando o contexto enfatizado acima, esta pesquisa tem como motivação melhorar a qualidade dos projetos de *software*, através do aprimoramento na etapa de elicitação de requisitos de software em uma organização específica, com o intuito de constatar as possíveis inconsistências no processo, visando reduzir os custos de manutenção, aumentar a satisfação dos clientes e melhorar o prazo de entrega dos projetos.

A pesquisa relatada neste artigo busca responder a seguinte questão norteadora de pesquisa: Como a elicitação de requisitos baseada em métodos ágeis pode aprimorar a qualidade do software na organização XYZ?

Deste modo, o objetivo geral da pesquisa constitui-se em: propor um artefato guia para a realização da elicitação dos requisitos pelo/junto ao cliente.

A ideia principal é que tal artefato venha a ser utilizado, se possuir boa adesão, como exemplo para implementação de um assistente de elicitação ao sistema interno da organização, atualmente chamado de "Service Desk".

Tendo por base o objetivo geral, definiu-se os seguintes objetivos específicos: 1) identificar junto à empresa as técnicas e métodos utilizados atualmente para a elicitação de requisitos, mapeando os pontos fortes e os pontos fracos; 2) analisar o método atual da empresa do estudo de caso quanto à eficácia; 3) propor um artefato para elicitação de requisitos baseado em métodos ágeis; 4) avaliar a melhoria nesta etapa junto aos gerentes do setor de tecnologia da informação (TI) frente ao uso do artefato.

Este artigo está organizado em seis seções, incluindo Introdução e Conclusão. A Seção 2 apresenta o referencial teórico empregado na elaboração do trabalho, conceituando requisitos e os métodos de elicitação, tanto no método tradicional quanto em métodos ágeis. A Seção 3 apresenta um resumo de trabalhos relacionados à elicitação de requisitos e em que eles se diferenciam da presente pesquisa. Na Seção 4 é apresentada

a metodologia utilizada no desenvolvimento do estudo. A elaboração do estudo na pesquisa se encontra na Seção 5. E a Seção 6, apresenta as conclusões do autor.

2. Requisitos

Nesta seção, será abordado o conceito de requisitos de *software*, ao mesmo passo que serão apresentadas as principais técnicas de elicitação, algumas técnicas de elicitação em métodos ágeis e trabalhos relacionados com o estudo.

2.1 Definição

Koscianski (2007), define requisitos de *software* como a transcrição do comportamento que o *software* deve apresentar quando estiver pronto para ser executado, ou seja, funções e detalhamento das operações que devam realizar, bem como o detalhamento das suas propriedades ou atributos.

De mesmo modo, Sommerville (2011) apresenta os requisitos de um *software* como a definição do que este deve fazer, os serviços que irá oferecer e as limitações de sua atividade. E, resume os requisitos como a representação das necessidades dos clientes para um *software* que possui uma finalidade específica, como por exemplo: o controle de um equipamento, registro de um pedido ou busca por informações.

Pressman (2011), apresenta os requisitos de forma semelhante à supracitada, ou seja, são o que o cliente deseja, suas necessidades para com o *software*, as especificações dos recursos que o *software* apresentará ao ser chamado de produto.

Segundo Souza (2011), os requisitos de um *software* são a base para a sua implementação, pois são as definições dos principais recursos de um produto de *software*, isto é, são as funcionalidades e as restrições que deve possuir.

Diversos autores classificam os requisitos de forma a propor uma organização em um documento de requisitos e facilitar o entendimento de acordo com o leitor, cada um a seu modo. Sommerville (2011) e Koscianski (2007) os classificaram em requisitos de usuário e de sistema.

- Requisitos de usuário: são declarações, em linguagem natural com auxílio de representações, dos serviços que o sistema fornecerá aos clientes e suas restrições com as quais estes devem lidar;
- Requisitos de sistema: são os detalhamentos, tanto das funções, como dos serviços e das restrições operacionais do *software*. O artefato de requisitos tem o papel imprescindível de definir pontualmente o que deve ser implementado.

Classicamente, subdivide-se estes requisitos em mais duas categorias: funcionais e não-funcionais, tanto para os requisitos de usuário, quanto para os requisitos de sistema [SOMMERVILLE 2011]:

- Requisitos funcionais: Descrevem as funcionalidades que se espera que o sistema forneça, como deve reagir às entradas e qual o tratamento utilizado em determinadas situações. Opcionalmente, os requisitos funcionais podem descrever o que o sistema não deve fazer;
- Requisitos não-funcionais: São restrições ao sistema como um todo. Incluem restrições de desempenho, restrições no processo de desenvolvimento e restrições impostas por normas.

2.1.1 Requisitos funcionais

Conforme mencionado no item anterior, estes requisitos descrevem as funcionalidades esperadas ao final do desenvolvimento do produto de *software*, como entradas, saídas, exceções, entre outras [KOSCIANSKI 2007].

Quando categorizados como requisitos de usuário, usualmente são descritos de forma abstrata, com o objetivo de facilitar a compreensão dos usuários do sistema. Em contrapartida, requisitos de sistema funcionais descrevem em detalhes as suas funções, suas entradas e saídas, exceções, entre outras [SOMMERVILLE 2011].

De todo modo, os requisitos funcionais devem ser especificados detalhadamente, de forma consistente e completa, todas as funções desejadas pelo cliente deverão estar descritas de forma não contraditória. Entretanto, quanto mais complexo for o *software*, mais difícil se torna atingir a completude dos requisitos. O principal motivo para isso acontecer, é o fato da equipe possuir diferentes pontos de vista sobre os mesmos requisitos [KOSCIANSKI 2007].

Muitos requisitos irão depender do tipo do *software* a desenvolver, do que os usuários desejam e das necessidades da empresa. Elencados a seguir, encontram-se alguns exemplos de requisitos funcionais [SOUZA, 2011]:

- Cadastro de fornecedores;
- Relatório de compras;
- Pesquisa de materiais;
- Escolha de materiais;
- “*Seleção da forma de pagamento*”.

2.1.2 Requisitos não-funcionais

Como o próprio nome nos instiga a pensar, requisitos não-funcionais não possuem relação direta com os recursos oferecidos pelo *software* aos seus usuários. Podem estar relacionados às propriedades resultantes do sistema, como desempenho, ocupação de área em disco, confiabilidade, entre outras [SOMMERVILLE 2011].

Koscianski (2007) cita dois exemplos de requisitos não funcionais. O primeiro se refere a um *software* para controle de voo, mesmo que todos os recursos tenham sido implementados, se o sistema não possuir uma boa confiabilidade, não poderá ser utilizado. E o segundo, cita um sistema de terminal bancário, onde expressa o problema desempenho, pois se o sistema demorar uns 10 minutos para gerar apenas o saldo, estará fadado ao fracasso.

Como pôde ser notado no texto apresentado acima, deixar de atender um requisito não-funcional pode resultar na inutilização de todo o sistema [SOMMERVILLE 2011]. Souza (2011), por sua vez, destaca que há diversas propostas de classificação que visam organizar os requisitos não funcionais. Entretanto, afirma que cabe a cada organização definir a categorização a ser utilizada para padronizar a especificação de requisitos.

Os requisitos não funcionais, de acordo com a proposta de Sommerville (2011), podem ser classificados em três categorias iniciais: requisitos de produto, requisitos de processo e requisitos externos.

1. Requisitos de produto: são requisitos que especificam ou restringem o que o *software* deverá fazer;
2. Requisitos organizacionais: são os requisitos gerais de sistemas, oriundos das políticas e procedimentos da organização do cliente e do desenvolvedor;

3. Requisitos externos: são todos os requisitos que provém de fatores externos ao sistema e seu processo de desenvolvimento.

Estas categorias por sua vez, se subdividem em outras, conforme representado na figura 1.

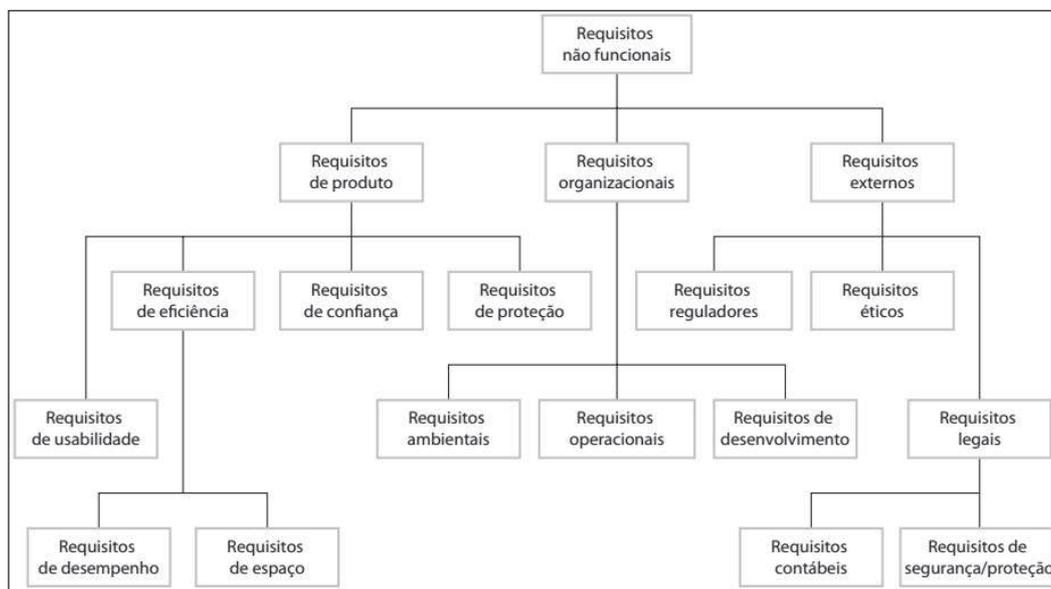


Figura 1 - Classificação de requisitos não-funcionais

Fonte: Sommerville, 2011.

Como reforça Sommerville (2011), propor requisitos não funcionais como metas gerais estabelecem boas intenções, mas são uma porta de acesso a problemas, pois deixam margem para interpretação dos desenvolvedores e, tão logo, para disputas quanto à entrega do sistema.

Preferencialmente, os requisitos não funcionais precisam ser quantitativos, pois é necessário que sejam testáveis. Pode-se mensurar essas características na fase de testes do sistema, com o intuito de verificar se ele tem cumprido ou não seus requisitos não funcionais. No entanto, os clientes de um sistema consideram difícil traduzir suas metas em requisitos mensuráveis. Existem algumas metas como manutenibilidade por exemplo, que não há uma métrica que possa ser utilizada. Mas também há casos que, mesmo quando é possível especificar quantitativamente, dificilmente os clientes conseguem relacionar suas necessidades com as especificações. Eles não compreendem o significado de um número definindo a confiabilidade necessária, pois é mais uma terminologia utilizada em sistemas computacionais. E como o custo para averiguar requisitos não funcionais mensuráveis pode tornar-se muito elevado, é difícil justificar os custos para os clientes, os quais pagam pelo sistema [SOMMERVILLE 2011].

Koscianski (2007) afirma que é frequente a existência de conflitos entre requisitos funcionais e não funcionais, como por exemplo: caso seja limitado o tamanho em bytes de uma aplicação embarcada em um dispositivo móvel, poderá se tornar incongruente para com o número de funções a serem implementadas.

Muitas vezes é difícil fazer uma distinção clara entre requisitos funcionais e não funcionais. Sua classificação dependerá principalmente do seu nível de detalhamento, destaca Souza (2011).

Para reforçar sua afirmação, Souza (2011) cita o seguinte exemplo:

“Se definirmos que o sistema deve garantir que os dados sejam acessados apenas por pessoas autorizadas, estamos definindo um requisito não funcional (RNF), pois trata-se de uma característica do sistema como um todo e não de uma funcionalidade. Entretanto, se definirmos que o sistema devem possuir uma funcionalidade para autenticação na qual os usuários devem se identificar através de um nome de usuário e senha, estamos definindo uma função a ser disponibilizada pelo sistema e, neste caso, um requisito funcional (RF)” [SOUZA 2011].

De certo modo, um requisito não funcional mais abstrato, ser decomposto em requisitos funcionais mais detalhados é fácil de se encontrar. Ao ponto que, para atender a um requisito não funcional, vários requisitos funcionais poderão ser necessários [SOUZA 2011].

Para Sommerville (2011), o processo de engenharia de requisitos de *software* pode subdividir-se em quatro atividades primordiais:

1. Estudo de viabilidade: é definido pela construção de um sistema objetivando à rentabilidade do negócio;
2. Elicitação e análise: caracteriza-se pela identificação das necessidades do sistema a ser construído;
3. Especificação: é a fase em que se descreve todas as informações obtidas na etapa de elicitação;
4. Validação de requisitos: consiste na averiguação da abstração do sistema, pois confere se a mesma está associada à realidade, se os requisitos são consistentes, completos, válidos e verificáveis.

A partir da seção seguinte, será detalhado o conceito de elicitação e análise de requisitos e, serão apresentados alguns métodos para a execução deste importante e complexo passo na engenharia de requisitos.

2.2 Elicitação

“The hardest single part of building a System is deciding what to build... No other part of the work so cripples the resulting of a system if done wrong. No other part is more difficult to rectify later” [BROOKS, 1987].

Conforme a citação anterior de Brooks (1987), a atividade mais desgastante no desenvolvimento de um *software* consiste na identificação do que construir. Não há outra atividade do projeto que comprometa tanto o resultado final se elaborado de forma errônea. Nenhuma outra atividade confere tanta dificuldade em correções posteriores.

A elicitação de requisitos, segundo Souza (2011), define-se por um conjunto de atividades associadas à exploração dos requisitos de um sistema onde, em conjunto, os analistas, clientes e usuários finais trabalham para compreender as necessidades do sistema, valendo-se do detalhamento de todas as funcionalidades e de suas restrições. De modo similar, Pressmann (2011) descreve a elicitação de requisitos como um trabalho em equipe, pois os interessados unem forças para identificar e entender o problema, pensar nas soluções, negociar diferentes abordagens e em conjunto, elencar os requisitos preliminares, isto é, a combinação das atividades de solução de problemas, elaboração,

negociação e especificação. Para Sommerville (2011), a elicitação de requisitos consiste em informações resultantes da união dos conjuntos de dados do sistema requerido e dos sistemas existentes, ou seja, faz-se uma análise do que é requerido e o que já está sendo atendido pelos sistemas atuais. Para gerir estas informações, inclui-se documentação, pessoas envolvidas na operação do sistema (conhecidas como *stakeholders*) e especificações técnicas de sistemas similares. De modo a colaborar com os *stakeholders*, provendo-lhes uma ideia do que esperar do sistema, pode-se fazer uso de cenários e protótipos criados a partir de interações.

De acordo com Pressmann (2011), abordagens colaborativas semelhantes à supracitada, realizam o levantamento de requisitos em esferas parcialmente diferentes, contudo, todas se valem de variações das orientações básicas a seguir:

- Realizar reuniões com a participação dos engenheiros de *software* e outros interessados;
- Estipular regras para elaboração e participação;
- Sugerir uma agenda pouco formal, mas que cubra os pontos importantes sem afetar a fluidez das ideias;
- Definir um “facilitador”, das partes interessadas, para conduzir reuniões;
- Valer-se de ferramentas para definição dos requisitos, como por exemplo: planilhas, adesivos de parede, fóruns, *flip charts*², etc.

“A elicitação de requisitos não se trata apenas de questionar os clientes e usuários sobre o que eles precisam, mas sim de uma análise da empresa, do domínio da aplicação e dos processos de negócio envolvidos” [SOUZA 2011, cap. 5].

Idealmente, uma elicitação de requisitos moldável à realidade de cada empresa, seria mais facilmente empregada, no entanto, muitos desenvolvedores, usuários e gerentes não compreendem a necessidade da especificação de requisitos, pois pensam apenas em atender a demanda o quanto antes. Um dos principais motivos para isto, é o fato do tempo e dinheiro investido para uma boa especificação, porém, a história nos apresenta fatos que argumentam contra esta crença, pois sem uma boa especificação de requisitos, o *software* a ser desenvolvido custará ainda mais tempo e dinheiro [KOSCIANSKI, 2007].

Souza (2011) enfatiza que a identificação dos requisitos apenas aparenta ser descomplicada, pois envolve muitas pessoas e fatores abstratos que enredam seus resultados, e, elenca os problemas a seguir como principais fatores do inadequado levantamento de requisitos:

- Informações a respeito do âmbito da aplicação dispersas em textos, manuais e no conhecimento intrínseco às pessoas que atuam na área;
- As partes que compreendem o problema a ser resolvido, ou estão ocupadas tentando solucioná-lo, ou não querem investir seu tempo na colaboração com o analista;
- Interesses e políticas internas discrepantes;

² Termo em Inglês para um bloco de papéis preso a um tripé ou cavalete, com intuito de agilizar reuniões, pois não é necessário apagar o conteúdo escrito, apenas virar a página.

- Os clientes ainda não sabem o que querem, ou não conseguem explicar suas necessidades;
- Durante ou após a elicitação, ocorrem mudanças no negócio;
- Troca dos colaboradores envolvidos no processo.

Conforme representado pela Figura 2, há quatro dimensões na etapa de elicitação de requisitos que deve-se considerar [SOUZA, 2011].



Figura 2 - Dimensões da etapa de elicitação de requisitos

Fonte: Kotonya e Sommerville, 1998, apud Souza, 2011.

A dimensão identificada como “domínio da aplicação”, diz respeito à compreensão da área de negócio para a qual o sistema será proposto. O “problema a ser resolvido”, consiste em um estudo detalhado a respeito do problema específico, mencionado pelo cliente e, para o qual o sistema será implementado. As “necessidades do cliente”, remetem à uma compreensão detalhada das necessidades específicas do cliente, visando um sistema que apoie e some em seu trabalho. O “contexto do negócio” trata do entendimento de como o sistema irá interferir no negócio do cliente, ou seja, se irá contribuir para os objetivos do negócio. Deste modo, um sistema é aceito dependendo do quanto ele atende as necessidades e expectativas de seus utilizadores [SOUZA, 2011].

A Figura 3 a seguir, representa um modelo geral da etapa de elicitação e análise de requisitos conceituado por Sommerville (2011), segundo ele, cada organização deterá sua versão deste modelo, o qual irá variar de acordo com o *know-how*³ de seus colaboradores, a característica do sistema requisitado, normas, entre outros.

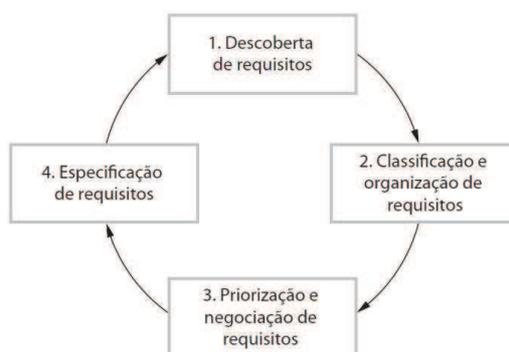


Figura 3 - O processo de elicitação e análise de requisitos

Fonte: Sommerville, 2011.

³ Refere-se ao conhecimento específico na realização ou solução de uma tarefa/problema prático, isto é, a eficácia de realização de alguma tarefa utilizando o conhecimento adquirido.

A primeira atividade do processo de elicitação e análise de requisitos, descrita como descoberta de requisitos, tem por finalidade a interação com os *stakeholders* do sistema, objetivando elencar se produto, ou seja, os requisitos. Também é nesta atividade que são revelados os requisitos de domínio dos *stakeholders*, como novos requisitos funcionais, determinação de formas de cálculos específicos, restrições dos requisitos funcionais, entre outros, bem como os requisitos da documentação [SOMMERVILLE 2011].

Logo em seguida, a atividade de classificação e organização dos requisitos detém a coleção de requisitos não estruturados e os arranja de modo a atingir uma congruência com a arquitetura do sistema, pois de modo usual, utiliza-se modelos da arquitetura do sistema para relacionar os requisitos a cada subsistema. Para o êxito desta atividade, a engenharia de requisitos e o projeto da arquitetura devem ser executados visando o objetivo de ambos [SOMMERVILLE 2011].

Após, entra em cena a atividade de priorização e negociação de requisitos, pois durante a fase de elicitação entre os vários *stakeholders*, alguns requisitos entram em conflito e, é nesta etapa que realiza-se a priorização e a resolução dos conflitos gerados durante a negociação de requisitos. Para que isso ocorra, os *stakeholders* necessitam marcar reuniões com intuito de resolver suas diferenças para chegar a um acordo satisfatório [SOMMERVILLE 2011].

A resolução destes conflitos se faz imprescindível para evitar-se decisões unilaterais, pois conforme comentado por Yourdon e Weingerg, apud Koscianski, (2007), frequentemente ocorre o fato de desenvolvedores assumirem o papel de cliente e estabelecer requisitos, considerados por estes como os “ideais” para o produto. Ou ainda, fatores políticos, como *stakeholders* de maior influência estipularem requisitos que os beneficiem de forma particular.

Como última atividade, a especificação dos requisitos tem o objetivo de documentar e inserir os requisitos no próximo ciclo do processo. Durante esta atividade, podem ser produzidos documentos formais ou informais de requisitos [SOMMERVILLE 2011].

A elicitação de requisitos ideal, consiste em um processo cooperativo, para que deste modo, a especificação seja a mais fiel possível à realidade. O comprometimento dos *stakeholders* é de suma importância, pois durante este processo, alguns percalços aparecerão para dificultar o relacionamento entre os envolvidos, como a utilização da discussão de requisitos para evidenciar poder ou ressentimentos na equipe. Cabe ao gerente do projeto estar atento à estas situações e, encontrar meios de controlá-las [KOSCIANSKI, 2007].

Considera-se necessário levar em consideração a preparação dos desenvolvedores, pois estes não se preparam corretamente para efetuar uma boa elicitação de requisitos, e muitas vezes desconhecem o negócio em que o *software* será aplicado e suas regras, pois para um profissional formado em computação, compreender certas características de negócios torna-se uma tarefa árdua, o que exalta a necessidade de incluir o cliente nesse processo. A utilização de diagramas de casos de uso e/ou diagramas de classes ao discutir os requisitos com o cliente, faz com que o mesmo tome conhecimento do trabalho envolvido e as dificuldades encontradas, bem como as consequências de uma mudança [KOSCIANSKI, 2007].

No final, o desenvolvimento de um novo sistema, mesmo que seguindo rigorosamente os manuais e políticas organizacionais bem definidas, as quais indicam como os colaboradores devem executar suas tarefas, gerará mudanças no processo, podendo ocasionar alterações na forma de trabalho e, como alguns colaboradores podem ter encontrado meios mais produtivos e não os descreveram nestes manuais, podem não encontra-las no novo sistema, gerando resistência à mudança [KOSCIANSKI, 2007].

O tratamento destas adversidades envolve a aplicação, por parte dos desenvolvedores, de algumas técnicas de elicitação de requisitos que serão descritas na seção seguinte.

2.3 Técnicas de Elicitação

A técnica utilizada de modo mais usual para a elicitação dos requisitos é a entrevista. No entanto, para elencar-se o maior número possível de requisitos, essa técnica não é suficiente. Deste modo, cabe ao analista de sistemas ter o conhecimento pertinente das diferentes técnicas de elicitação existentes e saber como e quando aplicar cada uma delas. A definição das técnicas e o meio de integração estão diretamente relacionados com o problema e a equipe do projeto, pois conhecer as técnicas e saber identificar em que uma técnica é superior a outra, é o mais importante [SOUZA, 2011].

A presente seção descreve as principais técnicas de elicitação de requisitos de *software*, são elas: Entrevistas, Leitura de documentos, Observação, Questionários, Análise de protocolos, Participação ativa dos usuários, Reutilização de requisitos, Cenários, Casos de uso e Prototipação.

2.3.1 Entrevistas

É muito comum a aplicação de entrevistas para elicitar requisitos de *software*, pois através dela o analista obtém um entendimento do sistema sob diferentes pontos de vista. Sua principal vantagem consiste na proximidade para com o usuário, proporcionando rápida negociação, contudo, como há uma pluralidade de culturas e conhecimentos entre os usuários, muitos não conseguem expressar suas necessidades adequadamente, fazendo com que muitos requisitos não sejam identificados [SOUZA, 2011].

Pode-se realizar entrevistas do tipo abertas, sem questões predefinidas, entrevistas fechadas, com questões predefinidas e tutoriais, na qual o usuário apresenta a sua forma de trabalho, ou uma combinação destas. A qualidade da entrevista está diretamente ligada à abertura do analista para com os envolvidos, pois sempre que necessário, precisará mudar sua visão a respeito do sistema. Durante as entrevistas, cabe ao analista nortear as questões pertinentes aos requisitos e/ou discussão sobre sistemas utilizados [SOUZA, 2011]. Iniciar uma entrevista com a pergunta "o que você precisa?", dificilmente resultará em informações úteis, mas com um contexto bem definido, torna-se muito mais fácil do que em termos gerais [SOMMERVILLE, 2011].

Pelo simples motivo dos usuários gostarem de falar a respeito de seu trabalho e dos problemas cotidianos, a técnica de entrevista se mostra muito eficiente para a elicitação de requisitos, entretanto, não é suficiente para elicitar todos eles, pois requisitos pertinentes ao domínio da aplicação dificilmente são tratados nesta técnica, devido aos usuários utilizarem termos específicos para explicar os processos executados diariamente e/ou, apresentam uma explicação falha, não convincente ou acham desnecessário explicar, visto que é óbvio para eles [SOUZA, 2011].

2.3.2 Leitura de documentos

Para melhor compreender o domínio da aplicação e a organização do cliente, é muito comum o analista se valer da técnica de leitura de documentos. Um estudo prévio da documentação a respeito da área de negócio para a qual o sistema deverá ser desenvolvido é muito importante, pois facilita a comunicação com os envolvidos. Estas informações podem ser obtidas em livros, sites e artigos da área, material do cliente, entre outros [SOUZA, 2011].

Devido à indisponibilidade dos usuários, os analistas podem utilizar documentos gerados/utilizados no dia a dia do usuário para leitura, como formulários, relatórios, tutoriais, etc. A leitura deste conteúdo tem o intuito de imergir o analista nos processos de trabalho para os quais o sistema será desenvolvido [SOUZA, 2011].

A principal vantagem da técnica de leitura de documentos, é o volume de informações que o analista consegue obter sem depender dos usuários, porém, é impossível compreender totalmente os processos apenas com a leitura destes documentos, pois muitos estão dispersos em vários locais, além de exigir um grande volume de trabalho, o que pode acarretar no comprometimento do prazo de entrega [SOUZA, 2011].

2.3.3 Observação

A melhor forma de entender o trabalho dos usuários, é observando-os enquanto executam suas tarefas, pois como é muito natural para eles, acham difícil fazer uma boa descrição do que fazem, ou quando a fazem, acabam omitindo dados [SOUZA, 2011].

Através da observação, o analista se insere do âmbito de trabalho do usuário e compreende como o mesmo é realizado. Esta técnica complementa as técnicas de entrevistas e análise de documentos, uma vez que há divergências entre as descrições formais e como as pessoas explicam os processos realizados [SOUZA, 2011].

A vantagem desta técnica está na obtenção de uma visão mais completa a respeito do contexto real de trabalho [SOUZA, 2011]. Esta visão inclui a descoberta de requisitos implícitos, os quais refletem os processos reais, ao invés dos processos formais [KOSCIANSKI, 2007].

Como desvantagem, é necessário investir muito tempo, pois é um processo pouco estruturado e não ideal para a descoberta de requisitos [SOUZA, 2011].

2.3.4 Questionários

Segundo Koscianski (2007), há momentos em que a distância impede a utilização das técnicas de entrevistas e observação. Neste caso, pode-se utilizar a técnica de questionários para elicitare os requisitos.

Este questionário é elaborado pelo analista, utilizando sua experiência, ele elenca questões em uma ordem lógica para que, com as respostas, consiga abstrair as necessidades dos envolvidos. Outra vantagem desta técnica, encontra-se no fato de se obter informações de maior qualidade dos envolvidos que se expressam melhor através da escrita, no entanto, a elaboração do questionário exige uma grande experiência do analista para identificar quais são questões ideais [KOSCIANSKI, 2007].

A técnica de questionários também pode ser utilizada em conjunto com a técnica de entrevista, pois através deste, o analista pode se preparar melhor para a entrevista, anotando as dúvidas para que durante a entrevista, sane-as [KOSCIANSKI, 2007].

2.3.5 Análise de protocolos

A técnica de análise de protocolo consiste em uma narrativa do trabalho pelo usuário, ou seja, através da explicação do que este último está fazendo, realiza-se uma análise com objetivo de elicitare fatos que dificilmente são observáveis, permitindo uma melhor compreensão [SOUZA, 2011].

A desvantagem se destaca na possibilidade do usuário não conseguir se expressar com clareza, além do fato de que poucos usuários conseguem explicar exatamente como uma tarefa é executada [SOUZA, 2011].

2.3.6 Participação ativa dos usuários

Com o objetivo de disponibilizar em tempo real a explicitação das dúvidas e a negociação das soluções propostas, incorpora-se na equipe do projeto, usuários-chave que possuam bastante experiência nos processos e conhecimento da empresa, ação pela qual a técnica recebe o nome de participação ativa dos usuários [SOUZA, 2011].

Sua desvantagem se encontra na necessidade de treinamento dos usuários e a falsa impressão da eficácia do sistema causada, pois decisões tomadas diretamente com um usuário-chave nem sempre refletem os interesses da organização [SOUZA, 2011].

2.3.7 Reutilização de requisitos

De acordo com Souza (2011), dentro de uma mesma área de sistema, apenas 15% dos requisitos de um novo projeto são exclusivos dele. A reutilização de requisitos, que geralmente é executada de modo informal, economiza tempo e esforço da organização, pois os requisitos existentes em outros projetos pertencentes à mesma área de sistema, já foram analisados e validados.

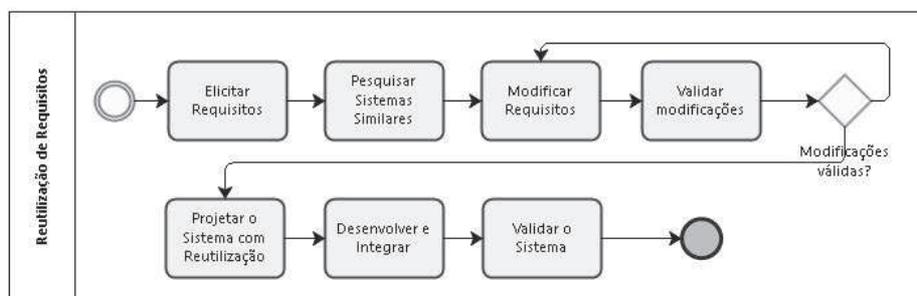


Figura 4 - Reutilização de requisitos.

Fonte: Elaborado pelo autor.

Apresentado na Figura 4, a existência de um processo padronizado de reutilização de requisitos em projetos, pode economizar muito esforço e aumentar a qualidade dos *softwares*. Devido a isso, enfatiza-se a importância na pesquisa de sistemas similares, sempre que um novo projeto for iniciado. A desvantagem desta técnica se mostra na dificuldade de se reutilizar requisitos sem necessidade de modificação, acarretando em uma nova validação dos requisitos modificados [SOUZA, 2011].

2.3.8 Cenários

De um modo geral, os envolvidos no projeto de um *software* consideram de mais fácil compreensão exemplos de situações reais, pois com eles o cliente pode se lembrar de

detalhes que não haviam sido revelados em questionários. Estes exemplos reais, chamados de casos de uso, constituem a técnica de cenários [KOSCIANSKI, 2007].

Os cenários podem ser confeccionados através de casos de uso da UML ou estórias de usuário. Este tipo de modelagem, trata o *software* como uma caixa preta, isto é, as informações internas do aplicativo não são relevantes [KOSCIANSKI, 2007].

Através destes cenários, o analista pode simular a execução das tarefas, representadas em alto nível. Através desta simulação é possível compreender melhor as funcionalidades esperadas para o sistema, assim como detectar exceções ou erros [KOSCIANSKI, 2007].

A Figura 5 exemplifica um cenário de autenticação de um sistema.

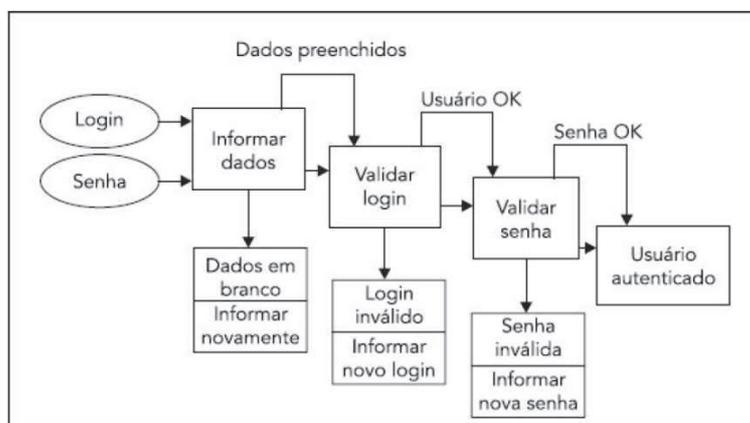


Figura 5 - Exemplo de um diagrama de cenário

Fonte: Souza, 2011.

“Como pode ser observado na Figura 5, o diagrama de cenário apresenta o fluxo normal de interação, além das informações de entrada, condições de mudança de estado e possibilidades de exceção.

No exemplo, o usuário deve informar seu login e senha para realizar o processo de autenticação. Uma primeira possível exceção é a tentativa de autenticar sem informar os dados necessários. No caso de o usuário informar os dados necessários, o sistema passa para a próxima etapa que é a validação do login. Nesta etapa, a exceção possível é o login estar incorreto e o usuário terá que informar novamente. Finalmente, na última etapa, caso o login seja validado, o sistema realiza a validação da senha. Como na etapa anterior, neste caso a exceção possível é a senha estar inválida e, neste caso, o usuário deve informar a senha novamente. Caso a senha seja válida, o usuário estará autenticado no sistema” [SOUZA, 2011].

2.3.9 Casos de uso

A utilização de casos de uso para descoberta de requisitos consiste na representação dos atores envolvidos em uma interação, e esta, recebe um nome e informações adicionais para descrever a interação com o sistema. Como informação adicional, pode-se incluir textos, modelos gráficos, diagramas de sequência ou estados da UML [SOMMERVILLE, 2011].

Os diagramas de casos de uso utilizam uma representação de alto nível, buscando atingir todas as possíveis interações que serão descritas como requisitos de sistema. Estes diagramas caracterizam-se por representar os atores como figuras “palito”, classes de interação como elipses, ligações entre os atores e interações através de linhas e indicação da direção da interação através de pontas de flechas, localizadas das extremidades das linhas de interação. Além da representação por figuras e formas geométricas, os casos de uso são acompanhados por uma descrição textual [SOMMERVILLE, 2011].

A Figura 6, apresenta um exemplo simples de um diagrama de casos de uso, que corresponde a um sistema para pesquisa e instalação de aplicativos em um *smartphone*. No exemplo corrente, o usuário do *smartphone* pode utilizar o sistema para sacar pesquisas por um aplicativo, obter informações sobre o aplicativo ou baixar e instalar o aplicativo em seu *smartphone*. Deste modo, temos o usuário como um ator e, as atividades pesquisar por aplicativo, obter informações do aplicativo e baixar aplicativo como casos de uso.

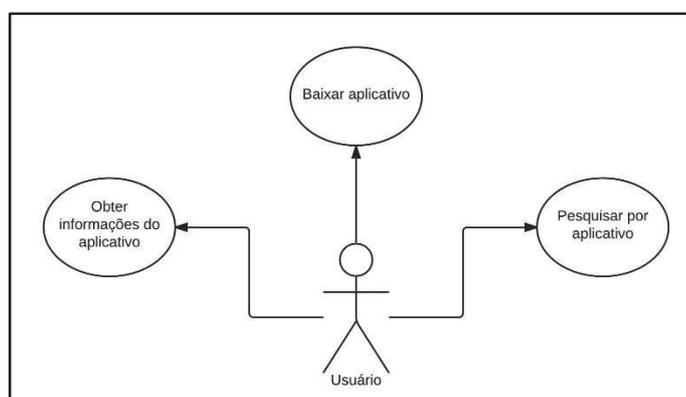


Figura 6 – Exemplo de um diagrama de caso de uso.

Fonte: Elaborado pelo autor.

Na modelagem de sistemas orientados a objetos, a linguagem UML tornou-se um padrão, desde então, casos de uso e elicitação baseada nestes, são extensivamente utilizados para a elicitação de requisitos [SOMMERVILLE, 2011].

2.3.10 Prototipação

Melhorar a compreensão da proposta do sistema pelo cliente e pela equipe de desenvolvimento, este é o objetivo da prototipação, a qual consiste na elaboração prévia de telas para o sistema, e estas, são incumbidas de representar visualmente como o sistema será disponibilizado ao usuário, facilitando o levantamento dos requisitos [SOUZA, 2011].

Essencialmente, segrega-se a prototipação em dois tipos distintos: descartável e reaproveitável. A prototipação descartável se mune de ferramentas que facilitam e agilizam o processo de criação de telas não funcionais para um sistema que está por ser desenvolvido, ou seja, estes protótipos servem apenas para esclarecimentos a respeito do sistema proposto. A prototipação reaproveitável, também consiste na elaboração prévia de telas do sistema proposto, entretanto, são confeccionadas como se o sistema estivesse em desenvolvimento, isto é, utiliza-se as ferramentas oficiais para produção das telas. Neste tipo de prototipação, tem-se a vantagem do tempo despendido, pois é relativamente menor em função do reaproveitamento [SOUZA, 2011].

Apesar de ser uma técnica muito eficaz na eliciação de requisitos, é necessário investir muito tempo para a prototipação, mesmo utilizando ferramentas de apoio, o que resulta na necessidade de aumentar o prazo do projeto [SOUZA, 2011].

A Figura 7 a seguir, apresenta um exemplo de um protótipo de tela para um sistema.

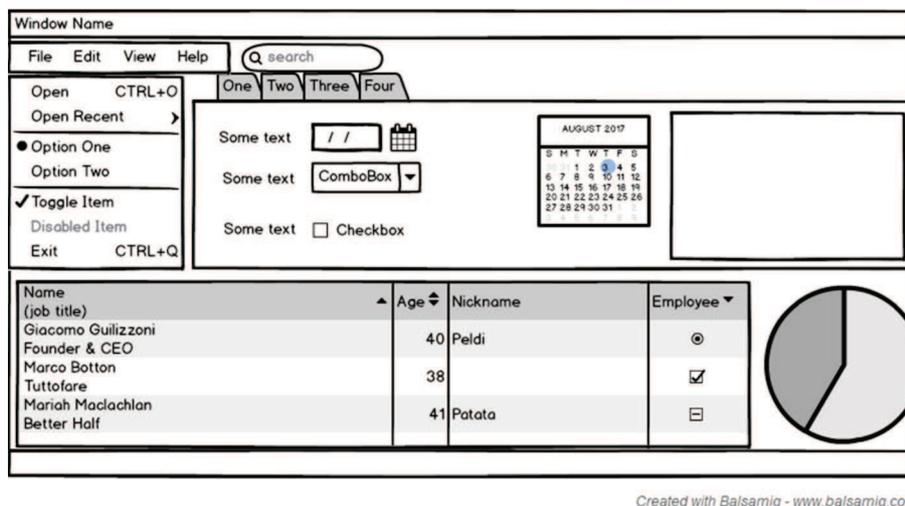


Figura 7 - Protótipo de tela de um sistema.

Fonte: Elaborado pelo autor.

2.4 Técnicas ágeis para eliciação de requisitos

As metodologias ágeis tornaram-se conhecidas a partir de 2001, quando dezessete especialistas em processos de desenvolvimento de *software*, os quais representam os métodos *Scrum* [Schwaber e Beedle (2002)], *Extreme Programming* (XP) [Beck (1999)] entre outros, decidiram estipular princípios comuns entre estes métodos.

Deste modo, firmou-se a Aliança Ágil, e com ela, foi criado o “Manifesto Ágil” [Agile Manifesto, 2001]. Dentre os princípios do “Manifesto Ágil”, Sommerville (2011) reforça a afirmação feita por este, e elenca os mais valorizados, são eles:

- Interações e os indivíduos são preferíveis a processos e ferramentas: A compreensão de como o *software* será utilizado pelos interessados e o que trará de benefício é de suma importância, pois de nada adianta um sistema sem erros se não será utilizado;
- *Software* funcionando, ao invés de extensa fundamentação: Deve-se priorizar o desenvolvimento do que trará real valor ao produto, pois investir muito tempo em detalhamentos aumenta o risco de não entregar o sistema funcionando;
- Cliente incorporado à equipe, no lugar da negociação de contratos: O mercado econômico é dinâmico, pois em projetos longos, as regras de negócio e as necessidades mudam. Deve-se lidar com as mudanças da melhor maneira possível;
- Atendimento rápido à mudanças, em oposição a planos rígidos: Coloque-se no lugar do cliente, pois quanto antes apresentar o sistema, mais rápido você terá o seu *feedback*. Deve-se investir no desenvolvimento dos requisitos preponderantes, entregando maior valor no menor prazo.

Com intuito de aprofundar o estudo a respeito dos métodos ágeis, nesta seção serão apresentados os principais métodos e, seus meios de elicitação.

2.4.1 Elicitação de requisitos com XP

Segundo Pressmann (2011), na programação XP, segue-se a orientação a objetos e, é baseada em quatro atividades primordiais: planejamento, projeto, codificação e testes. Dentre estas, concentra-se na atividade de planejamento a fase de elicitação de requisitos, a qual será abordada nesta subseção.

Também conhecida como "jogo do planejamento", a atividade de planejamento consiste em uma reunião da equipe XP onde, os integrantes procuram ouvir, isto é, buscam a compreensão do ambiente de negócios, das funcionalidades e dos fatores principais para o desenvolvimento do sistema [PRESSMANN, 2011].

Através desta atividade, o cliente cria histórias de usuário, as quais descrevem o resultado, as características e as necessidades do sistema. A cada história atribui-se um peso, sendo este determinado pelo próprio cliente, ou seja, uma prioridade em relação ao valor que a mesma possui para com o recurso ou função a ser desenvolvida. Cabe à equipe XP avaliar estas histórias afim de determinar uma estimativa de esforço, normalmente medida em semanas de desenvolvimento. Usualmente, se alguma história requerer mais que três semanas, é solicitado ao cliente um particionamento da história em histórias menores, repetindo-se as atividades de atribuição de estimativas [PRESSMANN, 2011].

A decisão de como unir estas histórias para a próxima versão do sistema, é determinada em conjunto pelos clientes e os desenvolvedores. Estes, por sua vez, buscam um consenso sobre quais histórias serão incluídas no projeto, qual a data para entrega, etc. A equipe XP utiliza três meios de ordenar as histórias: implementação imediata de todas histórias, priorização de implementação das histórias com maior valor ou priorização de implementação das histórias de maior risco [PRESSMANN, 2011].

Conforme as histórias são desenvolvidas, melhora-se a estimativa de desenvolvimento das próximas, pois é possível ter uma noção de quantas horas são necessárias para a implementação de cada uma. A entrega de histórias conforme a prioridade, caracteriza o desenvolvimento incremental, onde são produzidos os módulos do sistema conforme a necessidade do cliente. Este último, por sua vez, pode acrescentar histórias durante o processo de desenvolvimento, pode eliminar histórias, dividi-las ou muda-las conforme as prioridades do negócio. Neste caso, a equipe XP reavalia as mudanças e adapta o cronograma de acordo [PRESSMANN, 2011].

2.4.2 Elicitação de requisitos com Scrum

O Scrum consiste em um modelo ágil de processo que foi desenvolvido por Jeff Sutherland e sua equipe em meados da década de 1990, contudo, este método foca principalmente no desenvolvimento iterativo, ao invés de técnicas específicas da engenharia de *software* ágil [SOMMERVILLE, 2011]. A Figura 8 representa em um diagrama, o processo Scrum de gerenciamento.

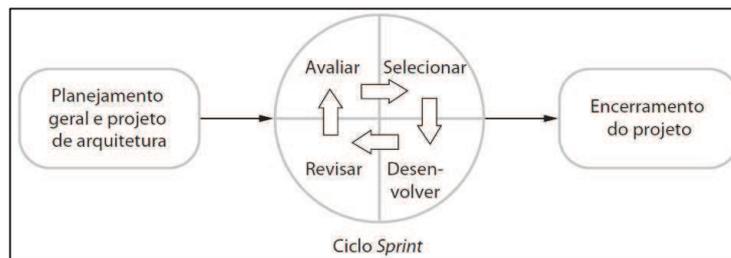


Figura 8 – O processo Scrum

Fonte: Sommerville, (2011).

A elicitação de requisitos no Scrum, está localizada na primeira das três fases que o compõem, denominada planejamento geral e, é nesta fase que se estabelecem os objetivos macros do projeto e da arquitetura do sistema, também conhecida como *product backlog* [SOMMERVILLE, 2011].

Entre as fases de planejamento e encerramento, encontra-se a parte vital do método Scrum, os ciclos *sprint* (períodos de algumas semanas para desenvolvimento de *releases*). Durante a fase de avaliação do *sprint*, revisa-se o *product backlog* para definir as prioridades para implementação e os riscos envolvidos. Nesta etapa, o cliente está intimamente envolvido, e é permitido a ele introduzir novos requisitos ou tarefas, sempre no início de cada *sprint* [SOMMERVILLE, 2011].

A equipe chegando a um consenso a respeito das funcionalidades, se organiza para iniciar o desenvolvimento do *software*. Com objetivo de analisar o progresso da equipe, são realizadas reuniões rápidas de acompanhamento e, quando necessário, reprioriza-se as funcionalidades [SOMMERVILLE, 2011].

Com intuito de evitar ruídos na comunicação durante a fase de desenvolvimento, as comunicações são mediadas pelo ‘*Scrum Master*’, o qual possui o papel de proteger a equipe de desenvolvimento de distrações externas [SOMMERVILLE, 2011].

Segundo Sommerville (2011), o Scrum incumbe à toda a equipe a tomada de decisões, evitando a utilização do termo ‘gerente de projeto’. Em contra partida, o ‘*Scrum Master*’ tem a função de organizar as reuniões diárias, controlar o *backlog* de trabalho, registrar decisões, medir o progresso comparado ao *backlog* e se comunicar com os clientes e a gerência externa.

2.4.3 Elicitação de requisitos com ASD

Proposto por Jim Highsmith, o desenvolvimento de software adaptativo (*Adaptive Software Development - ASD*) é uma técnica utilizada para desenvolvimento de sistemas complexos. Suas bases concentram-se na colaboração e auto-organização das equipes [HIGHSMITH, 1999].

Segundo Highsmith (1999), o desenvolvimento ágil e adaptativo baseado na colaboração consistem em “um recurso para organizar nossas complexas interações, tanto quanto disciplina e engenharia o são”. Diante disso, define o ciclo de vida do ASD através de três fases: especulação, colaboração e aprendizagem.

A elicitação de requisitos encontra-se na fase de especulação, cuja finalidade consiste na condução do planejamento de ciclos adaptativos. Este planejamento utiliza as informações iniciais do projeto, isto é, a definição da missão do cliente, as restrições do

projeto e os requisitos básicos para determinar o conjunto de ciclos de versão que serão necessários para o projeto.

Independentemente do quão completo um plano de ciclos pareça ser, sofrerá mudanças. Ao término de cada ciclo, utiliza-se suas informações para revisar o plano, ajustando-o de forma que o planejamento se adapte à realidade de trabalho da equipe ASD [HIGHSMITH, 1999].

A colaboração no método ASD é incentivada por Highsmith (1999) afim de multiplicar as habilidades e a capacidade criativa, tal colaboração é recorrente nos métodos ágeis. Entretanto, a colaboração entre os indivíduos não é algo trivial, pois necessita de comunicação e trabalho em equipe, o qual infere sobretudo uma questão de confiança entre os participantes.

2.4.4 Elicitação de requisitos com DSDM

O Método de Desenvolvimento de Sistemas Dinâmicos (*Dynamic System Development Method - DSDM*) é outra abordagem de desenvolvimento de *software* ágil que oferece meios para construir e manter sistemas que atendem restrições de prazo apertado, fazendo uso da prototipagem incremental [PRESSMANN, 2011].

Segundo Pressmann (2011), a ideologia do DSDM é baseada em uma adaptação do princípio de Pareto, onde é definido que 80% de uma aplicação pode ser entregue em 20% do tempo necessário para entrega da aplicação completa (100%).

Classificado como um processo iterativo, cada iteração obedece a regra dos 80%, isto é, deixa-se de lado detalhes em que os requisitos de negócio são desconhecidos, pois podem ser completados posteriormente, e direciona-se o foco na requisição do que é realmente necessário a cada incremento [PRESSMANN, 2011].

Nota-se que a elicitação de requisitos no DSDM está inserida nas duas atividades citadas por Pressmann (2011), o qual menciona que estas atividades precedem seus ciclos iterativos, são elas: a) estudo da viabilidade: atividade na qual se estabelece os requisitos básicos de negócio, restrições associadas e se avalia o projeto como um candidato ao processo DSDM; b) estudo do negócio: nesta atividade se define os requisitos funcionais e de informação que agregam valor de negócio à aplicação, define-se a arquitetura básica da aplicação e identifica-se os requisitos de fácil manutibilidade.

2.4.5 Elicitação de requisitos com FDD

Segundo Pressmann (2011), o desenvolvimento dirigido a funcionalidades (*Feature Driven Development - FDD*), foi concebido originalmente por Peter Coad e seus colegas como um modelo apropriado para a engenharia de *software* orientada a objetos. Posteriormente, Stephen Palmer e John Felsing aprimoraram o trabalho de Coad, apresentando um processo ágil adaptativo, o qual pode ser empregado em projetos de médio a grande porte.

Assim como outras metodologias ágeis, o FDD fomenta a colaboração entre as pessoas da equipe, gerencia problemas e complexidade de projetos através da segregação em funcionalidades e comunicação verbal, textual e gráfica para apresentação de detalhes técnicos. Encorajando o desenvolvimento incremental na concepção de projetos, o FDD enfatiza o controle de qualidade em um ciclo contínuo, através de inspeções do código e do projeto, aplicação de auditorias, coleta de métricas e utilização de padrões [PRESSMANN, 2011].

Basicamente, o FDD possui duas fases: a de Concepção e Planejamento e a de Construção. Na fase concepção e planejamento é que ocorre a triagem dos requisitos. Pode-se utilizar as técnicas tradicionais de elicitação dos requisitos, mas sem perder o foco das funcionalidades, pois a ênfase do FDD é justamente nestas últimas, as quais são passíveis de implementação em duas semanas ou menos [PRESSMANN, 2011].

3. Trabalhos Relacionados

Nesta seção são apresentados alguns trabalhos relacionados à elicitação de requisitos e desenvolvimento baseado em metodologias ágeis.

Batista (2003), em sua dissertação de mestrado, propôs uma sistemática de classificação das técnicas de elicitação de requisitos com o intuito de aumentar a fluidez da etapa de elicitação. Utilizando tabelas que comportam as características e particularidades de cada técnica, ele proporcionou ao desenvolvedor um meio para selecionar a que lhe parece mais aplicável. Entretanto, segundo Batista (2003), classificar uma técnica de elicitação através de suas particularidades é uma tarefa difícil, pois é muito subjetiva, além da possibilidade de geração de conflitos quanto à determinação das técnicas em termos.

Com a sistemática de Batista (2003), um engenheiro de requisitos pode comparar as características da técnica com o perfil da organização, escolhendo a que mais se encaixa. O ponto principal desta sistemática consiste na seleção de uma técnica de elicitação de requisitos através de critérios, e não apenas com a utilização da experiência do engenheiro.

Buscando despertar o interesse de alunos, Chiavegatti (2014) propõe o desenvolvimento de um jogo educacional para as etapas de elicitação e análise de requisitos, através da simulação de atividades práticas. Neste jogo, o aluno atua como um engenheiro de requisitos, realizando tarefas de coleta, organização e priorização dos requisitos em cenários simulados.

Segundo Chiavegatti (2014), este jogo visa auxiliar no ensino da engenharia de requisitos, mais especificamente, nas etapas de elicitação e análise de requisitos. Além disso, Chiavegatti (2014) espera que seu jogo auxilie os profissionais no que diz respeito ao conteúdo apresentado aos acadêmicos, contribuindo para um aprendizado prático da teoria, através da utilização de novas estratégias de ensino, tornando o ambiente de ensino mais agradável.

Gomes e Wanderley (2003) apresentam um artigo sobre o processo de elicitação de requisitos para a criação de *software* educativo. Neste artigo, eles apresentam as etapas que estão sendo utilizadas em projetos de aplicações educativas, conhecidas como (*learnware*).

Assim como argumentado por outros autores, Gomes e Wanderley (2003) citam que a parte mais difícil do projeto de sistemas educativos não é sua implementação, mas ter a informação do que será desenvolvido.

Expressando sua opinião a respeito dos projetistas, Gomes e Wanderley (2003) comentam que eles possuem uma opinião formada a respeito dos usuários e suas necessidades, contudo, esta opinião é incorreta e não se dão conta disso. Por este motivo, buscam apresentar em seu artigo um processo de engenharia de *software* educativo,

abordando principalmente o desenvolvimento de uma aplicação educativa, substituindo as decisões empíricas por especificações bem fundamentadas.

Segundo Gomes e Wanderley (2003), através deste novo processo, espera-se criar interfaces educativas de boa qualidade pedagógica, uma vez que as especificações são registradas de acordo com a representação do sistema, geralmente criada por projetistas e desenvolvedores, os quais incluem em suas representações o que acreditam que deva ser levado em consideração no projeto da interface.

Assim como Chiavegatti (2014) e Gomes e Wanderley (2003), Lima (2013) apresenta um trabalho de conclusão de curso voltado à educação, onde um processo de desenvolvimento de *software* educacional denominado PREÁ, visa contribuir utilizando boas práticas da Engenharia de *Software*, bem como das metodologias ágeis.

Neste trabalho, Lima (2013) comenta que enfrentou muitas dificuldades para desenvolver o processo PREÁ, visto que as bases de pesquisas especializadas no assunto são precárias, além da falta de especificação das técnicas, métodos e ferramentas. Preocupado com a qualidade e a educação, Lima (2013) enfatiza a importância dos processos padrões, que deem importância às técnicas de avaliação e validação de *software* educativo.

O processo PREÁ, de acordo com Lima (2013), consiste em uma adaptação dos métodos ágeis *Scrum* e *XP*, o qual utiliza práticas como: foco nos usuários, interações e planos de interação *Just In Time*. Seu principal objetivo é a gerência e controle das imprevisibilidades dos projetos de desenvolvimento educacional.

No artigo de Silva (2011), são apresentados os resultados de um mapeamento sistemático da literatura, compreendendo padrões de requisitos para escrita de requisitos de *software*. Neste mapeamento, Silva (2011) identificou 3 padrões distintos para escrita de requisitos, destacando três aspectos: uso de *template*, uso de padrões e relacionamento entre os padrões. Entretanto, não houve uma avaliação formal dos padrões de escrita de requisitos, pois os benefícios atribuídos ao uso dos padrões originam-se de avaliações informais.

Em sua dissertação de mestrado, Viera (2012) apresenta uma técnica de elicitação de requisitos orientada pela modelagem de processos de negócios. Esta técnica, denominada REMO (*Requirements Elicitation oriented by business process Modeling*), utiliza diagramas de processos de negócios para extrair os requisitos. Seu principal objetivo é auxiliar os analistas de sistemas a identificar os requisitos funcionais, não funcionais e regras de negócios a partir dos diagramas de processos de negócios.

Viera (2012) também descreve a forma de como a técnica REMO foi concebida e avaliada, bem como apresenta os resultados obtidos dos três estudos experimentais realizados, que indicam que a técnica REMO possa vir a ser útil no desenvolvimento de *software* durante a elicitação de requisitos.

Comparando os trabalhos supracitados à proposta da presente pesquisa, observa-se semelhanças com relação à engenharia de requisitos e sua etapa de elicitação e análise de requisitos, mais especificamente no quesito apresentação e utilização de técnicas de elicitação. Contudo, diferem-se no estudo realizado, visto que Batista (2003) elenca as técnicas de elicitação com intuito de classificá-las por características, objetivando auxiliar na escolha apropriada. Já Chiavegatti (2014), Gomes e Wanderley (2003), Lima (2013) e Silva (2011), ambos baseiam-se na literatura a respeito da elicitação de requisitos para

aprimorar de certo modo, técnicas para desenvolvimento de *software* educacional. Diferentemente, Vieira (2012) apresenta uma nova técnica para eliciação de requisitos, especializada em processos de negócios. Na presente pesquisa, pretende-se fazer uso de técnicas de eliciação de requisitos baseadas em métodos ágeis, as quais foram citadas na literatura, entretanto, em um sistema legado de mais de 20 anos, que utiliza linguagem de programação procedural e poucas ferramentas de apoio.

4. Materiais e Métodos

Esta seção apresenta o planejamento da pesquisa, a definição da amostragem que irá compor o trabalho, juntamente com as técnicas de coleta e análise dos dados, além de apresentar as limitações do estudo.

4.1 Delineamento da Pesquisa

A presente pesquisa possui natureza explanatória, o método de pesquisa consistem em estudo de caso único, tendo abordagem qualitativa. Segundo Yin(2001), questões de pesquisa do tipo "como" e "por que" caracterizam-se por traçar ao longo de um período, fatos históricos que comprovem não meras recorrências ou acontecimentos, mas que tanjam e/ou consigam embasar o processo através de um levantamento de dados documental ou através da projeção e condução de experimentos.

Segundo Gil (2010), não há um consenso entre os pesquisadores para a elaboração dos estudos de caso, isto é, não existe um roteiro rígido a ser seguido, contudo, percebe-se quatro fases que abrangem o seu delineamento:

1. Definição da unidade-caso: consiste na delimitação da unidade que representa o caso, sendo necessário habilidades do pesquisador para identificar quais informações são suficientes para compreender o objeto em sua totalidade;
2. Coleta de dados: caracteriza-se por procedimentos quantitativos e qualitativos, dentre eles podemos elencar: observação, análise de documentos, entrevistas formais ou informais, histórico de vida, aplicação de questionário com perguntas fechadas, levantamentos de dados, análise de conteúdo, etc. Existem diversos procedimentos que podem ser incorporados à esta lista;
3. Seleção, análise e interpretação dos dados coletados: consiste em uma fase conjunta, onde, a seleção dos dados deve imprescindivelmente considerar os objetivos da investigação, seus limites e um sistema de referências para avaliar quais dados terão utilidade ou não. Deverão ser analisados apenas os objetos que estiverem na esfera de dados selecionados, cabendo ao pesquisador definir antecipadamente seu plano de análise, considerando as limitações dos dados, principalmente no que se refere à qualidade da amostragem, pois com uma boa amostra, pode-se fazer generalizações a partir dos dados, caso contrário, deve-se apresentar os resultados em termos probabilísticos. A utilização de técnicas de análise derivadas de teorias reconhecidas no âmbito do conhecimento, evita que a interpretação dos dados implique em julgamentos implícitos, preconceitos, opiniões de senso comum, entre outros;
4. Apresentação dos resultados: é configurada pela elaboração dos relatórios parciais e finais, entretanto, deve-se especificar como foram coletados os dados; que teoria embasou a categorização destes e principalmente, quanto à sua veracidade. Estes relatórios devem preferencialmente ser sucintos, contudo, há situações em que um registro detalhado seja necessário, afim de explicitar os dados.

4.2 Unidade de análise

A empresa selecionada e apresentada neste estudo, por confidencialidade chamaremos XYZ, foi fundada em 1911, consistem em uma empresa metalúrgica 100% brasileira. Com a sede principal localizada na Serra Gaúcha do Rio Grande do Sul, é composta por 10 fábricas espalhadas pelo território nacional, onde, até o presente momento, emprega aproximadamente 7.000 pessoas, produz uma gama de produtos que gira em torno de 18 mil itens, sendo estes, direcionados ao segmento de linha “lar”, variando de utensílios domésticos à jardinagem, resultando na produção de mais de um milhão de peças por dia.

Para gerir tanta informação envolvida no processo de produção destes itens, a empresa XYZ possui seu próprio setor de TI. Localizado em sua sede, este setor é responsável pelo desenvolvimento de todo o ERP (*Enterprise Resource Planning*⁴) da empresa. Atualmente, este setor é composto por uma equipe 24 profissionais, onde a mesma se divide em 1 gerente de TI, 1 gerente de desenvolvimento e 22 desenvolvedores.

Como unidade de análise da pesquisa para o presente estudo, foi selecionado uma amostra não probabilística de profissionais da área de TI da empresa XYZ, os quais atuam como gerentes e outros, ora como analista de sistemas, ora como desenvolvedores, esta amostra é composta por dez profissionais selecionados por conveniência do pesquisador e, equitativamente, pela responsabilidade por eles exercida.

Através desta técnica, o pesquisador identifica os elementos aos quais possui acesso, admitindo que estes possam representar a amostragem. As principais vantagens da amostragem por conveniência ou acessibilidade são: custo baixo e rapidez (TORRES, 2000).

4.3 Técnicas de Coleta de Dados

A coleta de dados será realizada após a revisão bibliográfica, de modo quantitativo, através de informações provenientes de um software desenvolvido pela própria empresa XYZ, o qual será mencionado nesta pesquisa como "Service Desk" e, fornecerá dados do período de 1 ano.

4.4 Técnicas de Análise de Dados

A análise de dados terá como base a quantificação de alterações, as quais foram realizadas nos softwares desenvolvidos pela empresa XYZ. Será verificado a quantidade de retrabalhos que foram necessários nestes softwares com relação aos requisitos mal interpretados ou não informados. As conclusões desta análise serão de cunho comprobatório da necessidade de melhoria na etapa de elicitação dos requisitos dos softwares desenvolvidos, uma vez que esta etapa, é o objeto de estudo do trabalho.

4.5 Limitações do Método de Pesquisa

O método está limitado a um único estudo de caso, que irá propor uma melhoria no processo de desenvolvimento de software da empresa XYZ, definida e avaliada junto aos participantes, sem possibilidade de aplicação paralela à pesquisa, em função do curto prazo para a realização. Deste modo, as conclusões obtidas neste estudo não são generalizáveis, pois serão direcionadas especificamente para a melhoria na qualidade do

⁴ Sistema de informação que integra todos os dados e processos de uma organização em um único sistema. Fonte: Wikipedia.

software desenvolvido pelo setor de TI da empresa XYZ, com aplicação apenas em estudos posteriores.

4.6 Desenho da pesquisa

A Figura 9 a seguir, representa graficamente como foi concebida esta pesquisa no que tange aos passos metodológicos representados pelas etapas envolvidas.

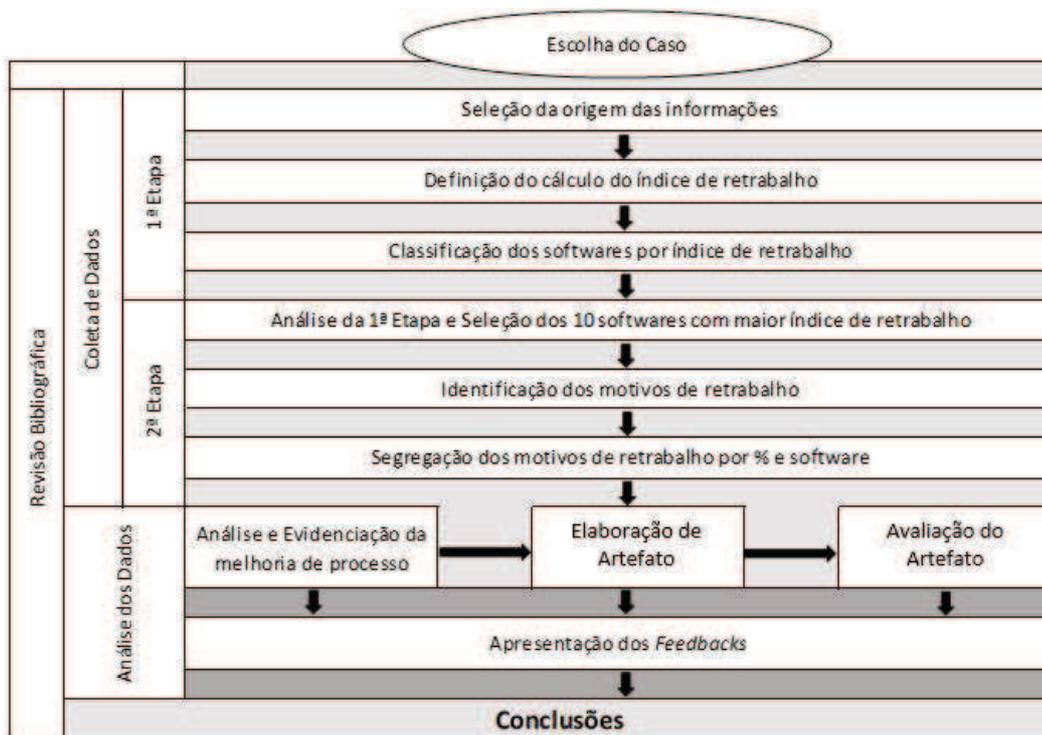


Figura 9 - Desenho da pesquisa.

Fonte: Elaborado pelo autor.

Toda a pesquisa se inicia com a escolha do caso, e, através deste, iniciou-se a revisão bibliográfica, na qual foi apresentada uma breve abordagem sobre requisitos de software. Logo após, o foco direcionou-se às atividades de elicitação de requisitos, utilizadas em métodos tradicionais e ágeis, apresentando a descrição destas e argumentando a importância do principal artefato produzido nesta fase: o documento de requisitos.

Para a coleta de dados, definiu-se duas etapas, as quais estão subdivididas em três fases distintas cada. Como primeira etapa, selecionou-se a origem das informações, que se refere ao material utilizado para a coleta de dados. Foi definido a fórmula para o cálculo do índice de retrabalho e a partir deste, classificou-se os softwares com maior índice de retrabalho no período de um ano. Na segunda etapa da coleta de dados, analisou-se a execução e os resultados da 1ª Etapa, após, foram selecionados os 10 softwares com maior índice de retrabalho para investigação. Primeiramente foi necessário identificar os motivos de retrabalho, e por último, segregou-se os motivos de retrabalho por índice em percentual e software.

Com as informações da coleta de dados, fez-se uma análise e evidenciação da melhoria da etapa de elicitação dos requisitos na empresa XYZ. Servindo de entrada para o processo seguinte, foi elaborado um artefato guia para aprimoramento no processo

supracitado. Tendo o artefato em mãos, foi realizada uma avaliação deste junto aos *stakeholders* chave, ou seja, gerentes e desenvolvedores *sênior*⁵ selecionados pelo pesquisador.

Após a avaliação do artefato proposto, foram apresentados neste trabalho os *feedbacks* dos *stakeholders* chave e as conclusões do pesquisador.

5. Estudo

Nesta seção, contextualiza-se a proposta de um artefato para aprimoramento da etapa de elicitação de requisitos, partindo da apresentação da unidade de análise, relatando o processo de desenvolvimento atual da empresa XYZ, passando para a etapa de Coleta Preliminar de Dados, responsável por identificar o cenário atual do levantamento dos requisitos na empresa XYZ, utilizada como base para a concepção do artefato proposto. A subseção 5.5 apresenta o detalhamento do desenvolvimento do artefato e na subseção 5.6 apresenta-se o *feedback* do artefato para elicitação de requisitos.

5.1 Contextualização da unidade de análise

De acordo com as informações supracitadas na subseção 4.2, o setor de TI da empresa XYZ desenvolve todo o sistema de gestão ERP⁶ da empresa internamente.

O desenvolvimento do ERP da empresa XYZ utiliza o modelo cascata como ciclo de vida, adaptado à sua realidade, bem como o modelo incremental. Dependendo da complexidade do sistema a ser desenvolvido, escolhe-se entre um modelo e outro. Até o presente momento, a equipe de desenvolvimento não adota metodologias ágeis para implementação de seus sistemas, e não possui certificação em níveis de maturidade, tanto para CMMI quanto para o MPS-BR.

Com objetivo de filtrar as demandas de manutenções e o desenvolvimento de novos recursos pela equipe de TI, firmaram-se os GTs (Grupos de Trabalhos), os quais são compostos por usuários chaves e subdivididos por sistema do ERP. Os GTs possuem uma agenda pré-definida, na qual os integrantes se reúnem para avaliar e discutir manutenções e/ou novos recursos para o sistema. Adicionalmente, participam destes encontros o gerente de desenvolvimento e o desenvolvedor sênior do respectivo sistema, afim de auxiliar nas dúvidas referentes à capacidade de implementação pelo TI e/ou tempo necessário que estes irão requerer.

Como o ERP desenvolvido pela empresa disponibiliza todo os meios necessários para as atividades da organização, tornou-se vital para o seu negócio. A Tabela 1 a seguir, apresenta algumas informações macro deste.

Tabela 1. Informações macro do ERP da empresa XYZ

	Quantidade
Usuários do sistema	2.205
Módulos	20
Programas	2.900

⁵ Termo em Inglês utilizado para indicar o indivíduo que acumula uma quantidade maior de anos de experiência em um mesmo cargo. Fonte: Wikipedia.

⁶ Do Inglês *Enterprise Resource Planning*, ou planejamento de recursos corporativo em Português, é um sistema de informação que integra os dados e processos de uma organização. Fonte: Wikipedia.

Fonte: Elaborado pelo autor

Se comparado à existência da empresa XYZ, a qual se faz presente a mais de uma geração na economia brasileira, seu ERP poderia ser considerado novo, com seus 20 anos de vida, entretanto, quando falamos em sistemas de informação, alguns anos podem fazer um sistema tornar-se obsoleto. O ERP da empresa XYZ consiste em um sistema legado, antigo, que vem recebendo manutenção a anos. Na época de sua concepção, não haviam ou não eram conhecidas as boas práticas de engenharia de *software*, e como resultado de sua idade avançada, as atividades de compreensão e manutenção deste sistema tornaram-se árduas.

5.2 A elicitação dos requisitos na empresa XYZ

De acordo com a Figura 10, a qual representa o processo macro de engenharia de *software* adotado pelo setor de TI da empresa XYZ, antes da fase de elicitação de requisitos, é preciso existir uma demanda de implementação ou manutenção. Esta demanda, é gerada/avaliada pelos GTs dos sistemas. Para realizar a validação de uma demanda, os GTs utilizam vários critérios de seleção, a seguir, alguns exemplos:

- Necessidade para o negócio da empresa: caracteriza a demanda como algo essencial para o sistema;
- Utilização: objetiva identificar se a demanda será utilizada constantemente ou se é necessária somente naquele momento;
- Retorno envolvido: implica na análise de retorno do investimento inferido à implementação;
- Recursos do TI para atendimento: visa determinar se o TI possui as ferramentas necessárias, ou se é preciso investir em novos recursos;
- Tempo para implementação: implícito no próprio nome, determina se o tempo existente é suficiente para implementar a demanda.

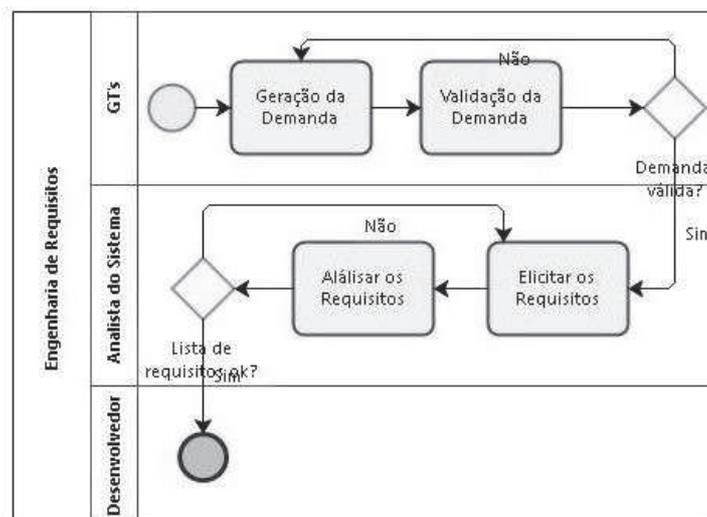


Figura 10 - Processo macro da engenharia de requisitos

Fonte: Elaborado pelo autor.

Após a validação da demanda, caso o GT a identifique como não válida, esta retorna à sua origem, para reavaliação ou cancelamento. Caso seja válida, é encaminhada

ao analista do sistema, que elicita e analisa os requisitos intrínsecos à demanda. Uma vez que a lista de requisitos esteja pronta, encaminha-se a demanda a um desenvolvedor que tenha conhecimento do sistema envolvido, caso contrário, se o analista entender que a demanda precisa ser amadurecida, ele a encaminha de volta ao GT. Entretanto, nem toda demanda passa pelos GTs, pois há demandas que dispõem este passo de análise, como por exemplo: governamentais, as quais se referem às mudanças em leis no país, econômicas, que refletem diretamente no faturamento, normativas, que implicam na certificação para fornecimento e conseqüentemente no faturamento, entre outras.

Estas demandas, muitas vezes chegam ao analista com uma situação de urgência inesperada, acarretando em uma elicitação de requisitos carente de detalhes, ocasionando uma má qualidade no *software*, pois devido ao curto espaço de tempo para implementação, acabam sendo mal codificados, com inserção de maus cheiros, códigos complexos e de difícil manutenção.

A documentação gerada acerca dos requisitos destas demandas é pobre, pois é baseada na elicitação gerada em pouco tempo, e apesar de ficar vinculada à solicitação do usuário, detalhes dos requisitos que são obtidos durante a implementação nem sempre são registrados, ficando apenas com o desenvolvedor o conhecimento das regras de negócio e particularidades envolvidas. Com isso, originam-se problemas de insipiência a respeito dos motivos que levaram o desenvolvedor à determinadas decisões no código.

Diante dos referidos fatos, o principal objetivo deste trabalho é realizar um estudo de caso para avaliar se as técnicas de elicitação de requisitos em métodos ágeis podem de fato, contribuir na melhoria da qualidade dos *softwares* do sistema.

5.3 Coleta preliminar de dados

Para realizar a coleta de dados, definiu-se a utilização de informações provenientes do *software* conhecido como “Service Desk” da empresa XYZ. Atualmente, o “Service Desk” é utilizado na empresa XYZ pelos seguintes integrantes:

- Usuários: colaboradores que utilizam o sistema diariamente nas operações dos processos da organização;
- Operadores: integrantes da equipe de TI descentralizada, que fazem o meio-campo entre usuários e desenvolvedores, normalmente uma ou duas pessoas por organização;
- Líderes dos GTs: usuário chave dentro de um GT, o qual é responsável por aprovações de demandas e organização dos encontros do grupo;
- Gerentes/Diretores: usuários de nível gerencial, responsáveis pela tomada de decisões que implicam diretamente no negócio da organização;
- Analistas de sistemas: responsáveis pela elicitação e análise dos requisitos;
- Desenvolvedores: responsáveis pela codificação e testes.

Tendo conhecimento de que o sistema “Service Desk” armazena suas informações em um *software* de banco de dados centralizado, denominado “Informix”, identificou-se as tabelas com as informações necessárias para a coleta de dados. Dentre as tabelas, duas são consideradas essenciais para a realização desta coleta: 1) tabela das solicitações: consiste na tabela principal para registro das demandas dos sistemas, 2) tabela dos objetos das solicitações: se refere aos sistemas criados/modificados durante o atendimento de uma demanda.

A coleta supracitada foi realizada utilizando uma linguagem de alto nível para banco de dados, denominada “SQL⁷”, esta linguagem é utilizada em larga escala para obtenção de informações dos mais variados bancos de dados.

Preliminarmente à coleta de dados, foram indagadas algumas questões sobre o que deveria ser considerado na coleta, resultando nos passos a serem seguidos. Essas questões estão dispostas a seguir:

1. Qual o período?
2. Quais objetos considerar?
3. Quais situações das demandas?
4. Quais os tipos de demandas?

Para o primeiro passo, determinou-se o período de coleta em um ano, iniciando em 01/07/2016 e fechando o período em 30/06/2017. O segundo passo foi determinar o tipo de objeto a pesquisar, onde definiu-se que para esta pesquisa, somente objetos do tipo “programa” seriam considerados, visto que, quando há manutenção em telas ou relatórios, o programa é recompilado e enviado às bases da organização. Quanto às situações das demandas, indagadas no terceiro passo, foi definido que seriam consideradas apenas demandas com situação “concluída”, pois demandas não concluídas ou canceladas não seriam relevantes para o presente estudo. No quarto passo, foi definido que apenas demandas do tipo “incidente” e “requisição” seriam consideradas, pois correspondem a correções ou implementações de novos recursos.

Conforme apresentado na Tabela 1, apurou-se para esta pesquisa que o ERP da empresa XYZ conta com 2900 programas, e destes, foi constatado que 64% necessitaram de alguma manutenção, fato evidenciado na Tabela 2.

Tabela 2. Manutenções realizadas no período

	Existentes	Com manutenção	(%)
Quantidade de programas	2900	1848	64%

Fonte: Elaborado pelo autor.

Visto que o objetivo desta pesquisa visa aprimorar a etapa de elicitação de requisitos no departamento de TI da empresa XYZ, foi necessário aprofundar as informações no quesito manutenção, afim de identificar os programas com maior incidência de erros, ou seja, retrabalho.

A Fórmula (1) a seguir, foi utilizada para o cálculo do índice de retrabalho, a qual foi definida como a soma dos incidentes de um programa, onde x representa o programa e n os seus incidentes:

$$x = \sum n \quad (1)$$

Estatisticamente seria mais interessante considerar o percentual de incidentes sobre as manutenções, entretanto, não se aplica ao presente estudo, pois muitos programas tiveram apenas duas ou três manutenções, as quais eram incidentes relativos à sua implementação, correspondendo a 100%, o que resultaria em uma amostragem não

⁷ Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional). Fonte: Wikipedia.

determinante. Assim, utilizou-se o quesito “número de incidentes”, pois quando há um grande número de incidentes, tem-se proporcionalmente um grande número de manutenções.

A Tabela 3 a seguir, nos apresenta um número surpreendente de incidência de erros, sendo notáveis 57% sobre o total de manutenções realizadas no período desta pesquisa. Ao verificar a quantidade de programas com mais de um incidente, o percentual baixa, entretanto, ainda é um número elevado, correspondendo a 33% das manutenções. Entretanto, somente com estes números não é possível apresentar evidências de problemas na etapa elicitação de requisitos.

Tabela 3. Número de incidentes nas manutenções

	Com manutenção	Com incidentes	Com mais de um incidente
Quantidade de programas	1848	1056	616
(%)	100%	57%	33%

Fonte: Elaborado pelo autor.

Afim de averiguar os motivos que levaram estes programas a possuírem um número tão elevado de incidentes, foram selecionados os dez programas com maior índice, os quais são apresentados na Tabela 4.

Tabela 4. Programas com maior índice de incidentes

Programa	Descrição	Manutenções	Incidentes	(%)
CMP095	Nota fiscal - livros fiscais	144	68	47
FTP061	Atualização do cadastro de notas fiscais	145	67	46
FVD011	Força de Vendas (Catálogo)	105	66	63
CDP341	Emissão das notas fiscais	119	62	52
DIP910	Geração do Sped Fiscal	194	56	29
CDP190	Atualização dos movimentos do CD	113	56	50
FTP361	Emissão das notas fiscais	105	53	50
FTP011	Atualização do cadastro de produtos	109	48	44
CMP080	Ordem de compra	89	45	51
GDD010	Gerenciador de documentos	70	45	64

Fonte: Elaborado pelo autor.

Após a identificação dos programas com maior número de incidentes, foi realizada a segregação dos motivos que levaram a incidentes. Nesta segregação, foram analisadas as demandas de incidente por programa, uma a uma verificando o conteúdo informado pelo usuário, e constatou-se quatro tipos distintos de incidentes, são eles:

1. Programação: erros relativos ao desenvolvimento do sistema;
2. Requisitos: erros em decorrência de falhas na elicitação dos requisitos;
3. Operacional: erros ocorridos por falha na execução do processo organizacional, isto é, o usuário realizou procedimentos que causaram o incidente;

4. Infraestrutura: erros onde o problema se origina no hardware, configuração, banco de dados ou comunicação.

Na Tabela 5 a seguir, são apresentados os números referentes aos incidentes do programas, onde cada um foi classificado conforme a segregação supracitada.

Tabela 5. Segregação de incidentes por programa

Origem do incidente	Programas									
	CMP09	FTP061	FVD01	CDP341	DIP910	CDP190	FTP361	FTP011	CMP08	GDD01
Programação	23	22	11	16	19	17	17	16	18	29
Requisitos	29	36	25	34	34	24	27	26	21	11
Operacional	15	7	4	11	3	10	9	5	6	1
Infraestrutura	1	2	26	1	0	4	0	1	0	4
TOTAL	68	67	66	62	56	55	53	48	45	45

Fonte: Elaborado pelo autor.

5.4 Constatando oportunidades de melhoria

À primeira vista, não está clara a evidência que levou o autor a realizar o estudo proposto, pois uma análise que utiliza apenas números, pode facilmente fazer com que uma informação importante passe despercebida, entretanto, façamos uma análise utilizando a Figura 11, que nos apresenta um gráfico com os números da Tabela 5.

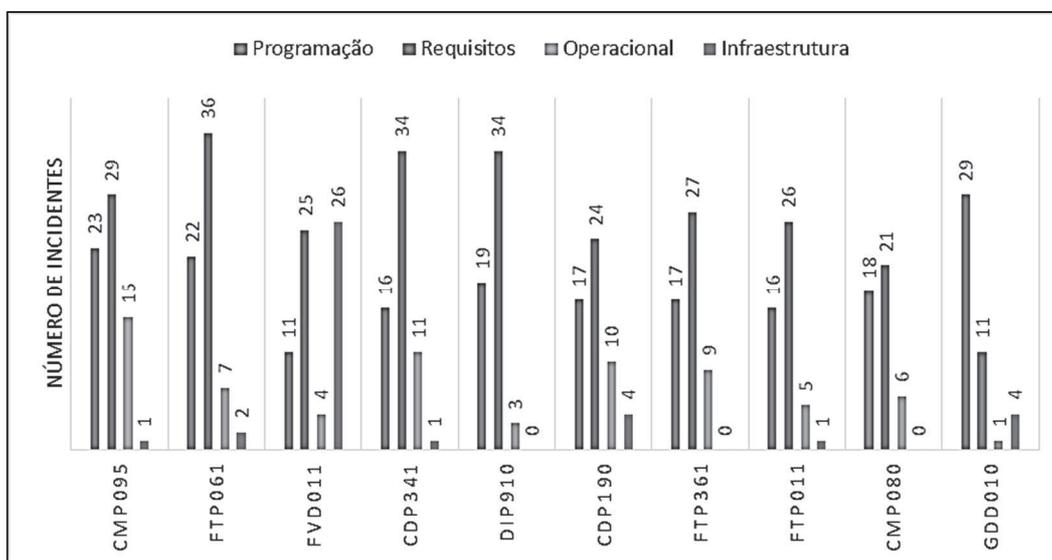


Figura 11 – Número de incidentes por tipo e programa

Fonte: Elaborado pelo autor.

De acordo com a legenda localizada na Figura 11, e com os números apresentados na Tabela 5, nota-se que a segunda coluna de cada programa, representa os incidentes por falha na etapa de elicitação de requisitos. Percebe-se que a coluna de incidentes com relação a requisitos mantém-se elevada em quase todos os programas analisados na coleta de dados. Esta constatação fica mais evidente na Figura 12 a seguir.

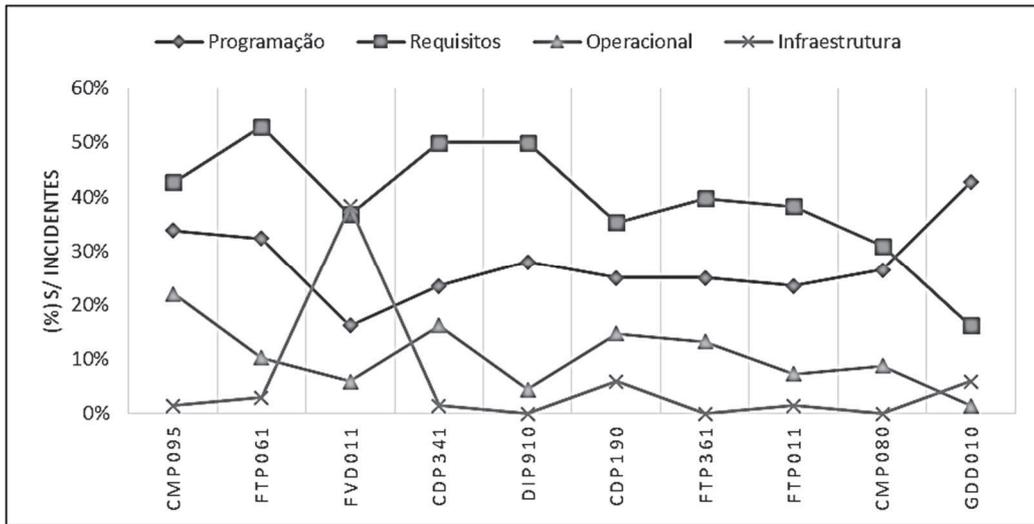


Figura 12 - Percentuais por tipo de incidente e programa

Fonte: Elaborado pelo autor.

A Figura 12 nos apresenta os dados coletados em um gráfico de linhas, dispondo os dados estatisticamente, os quais evidenciam coerentemente as premissas com relação ao estudo realizado, isto é, a maior incidência de erros concentram-se nos incidentes ocorridos devido à falhas na etapa de elicitação de requisitos.

Afim de reforçar esta constatação, na Figura 13 a seguir é apresentado um gráfico com o percentual de participação por tipo de incidente, com relação à soma dos incidentes ocorridos nos programas analisados.

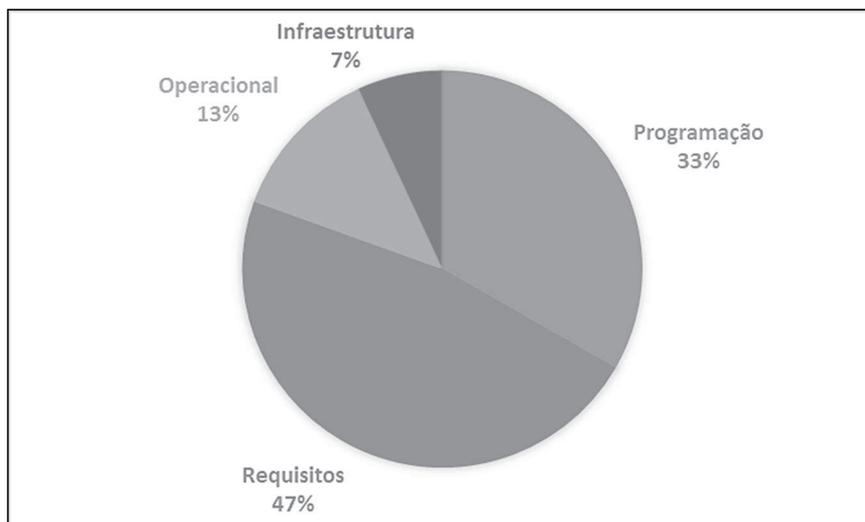


Figura 13 – Percentual de participação nos incidentes analisados

Fonte: Elaborado pelo autor.

Ao direcionar a atenção para a Figura 13, constata-se sem dúvida alguma, através da maior fatia de incidentes apresentada no gráfico, e das evidências supracitadas nas Figuras 11 e 12, que é incontestável a necessidade de um aprimoramento na etapa de elicitação de requisitos, pois esta é primordial na engenharia de requisitos.

5.5 Desenvolvimento do artefato para elicitación de requisitos

Com o objetivo de analisar o cenário atual de elicitación de requisitos da empresa XYZ, esta subseção apresentará uma análise da matriz *SWOT* (*Strengths, Weaknesses, Oppotunities and Threats*), que em Português significa (Forças, Fraquezas, Oportunidades e Ameaças). Segundo Machado (2005), esta técnica de análise de gestão foi desenvolvida por “Kenneth Andrews e Roland Christensen, dois professores da Harvard Business School” e, auxiliará na identificação dos pontos fortes e fracos na etapa de elicitación de requisitos, afim de prover dados para o desenvolvimento do artefato guia para aprimoramento da etapa de elicitación de requisitos na empresa XYZ.

De acordo com Machado (2005), a análise SWOT provê uma orientação estratégica significativa, pois permite às organizações:

- Sanar pontos fracos nas áreas em que a empresa enfrenta ameaças e tendências desfavoráveis;
- Identificar oportunidades a partir de seus pontos fortes;
- Corrigir pontos fracos nas áreas em que a empresa consegue visualizar oportunidades;
- Monitorar as áreas identificadas como pontos fortes, visando não ser surpreendida por possíveis riscos e/ou incertezas.

A seguir, são apresentados na Figura 14 os fatores e técnicas de elicitación de requisitos na matriz SWOT. Para os pontos fortes, foram elencados àqueles cuja constatação se faz presente na contextualização da unidade de análise e na descrição da elicitación de requisitos da empresa XYZ.

Análise SWOT	
Forças <ul style="list-style-type: none">- Comprometimento da equipe- Auto gerenciamento das equipes, maior autonomia, disciplina e regularidade- Respostas rápidas às demandas- Foco e maximização do ROI (Retorno do Investimento) e do Valor de Negócio- Participação ativa dos usuários- Habilidade técnica dos desenvolvedores- Condução de entrevistas- Produtividade- Transparência e visibilidade do status do projeto- Prototipação de sistemas	Oportunidades <ul style="list-style-type: none">- Treinamentos- Aquisição de software- Novas tecnologias
Fraquezas <ul style="list-style-type: none">- Tempo para desenvolvimento- Insipiência de usuários- Carência de detalhes na elicitación de requisitos- Falta de documentação da regra de negócio- Pouca flexibilidade para mudanças de requisitos- Entrega incremental apenas em módulos do sistema- Problemas de comunicação com o cliente- Falta de inspeção e adaptação do processo em busca da melhoria contínua e a redução dos desperdícios- Pouca antecipação de problemas e postergação da tomada de ações	Ameaças <ul style="list-style-type: none">- Desvalorização da moeda- Mudança na legislação- Escassez de mão-de-obra

Figura 14 – Matriz SWOT de fatores para elicitación de requisitos

Fonte: Elaborado pelo autor.

A determinação dos pontos fracos foi baseada nas características da elicitação de requisitos. A partir destas, avaliou-se quais podem ser consideradas como pontos fracos na empresa XYZ. Esta avaliação foi determinada de forma empírica pelo pesquisador, visto que o mesmo é colaborador da empresa.

Como oportunidades de melhoria, foram identificados e listados três fatores, que podem vir a contribuir com a elicitação de requisitos na empresa XYZ. E como ameaças, elencou-se três pontos que teriam capacidade de influenciar diretamente na equipe de TI da empresa XYZ, sem que esta tivesse controle.

5.5.1 Análise SWOT

Objetivando um aprofundamento da análise SWOT gráfica, foi determinada uma pontuação para cada fator em uma tabela (Apêndice I).

Conforme pode ser visualizado no Apêndice I, a primeira coluna possui os fatores que influenciam a elicitação de requisitos, os quais foram elencados e segregados conforme suas posições na matriz SWOT.

Na coluna “Peso”, deduzível pelo próprio nome, atribuiu-se um peso em percentual para cada fator, variando de 0 a 100 e baseando-se no provável impacto dessa característica sobre a estratégia de elicitação da empresa XYZ. A coluna “Classificação”, classifica os fatores através de valores que variam de 1 a 5, de fraco a excelente correspondentemente, baseando-se no atual estado em que o fator se encontra na empresa. Para determinar o resultado dos quadrantes, multiplicou-se o peso atribuído pelo valor de estado da classificação em cada fator, somando-os a cada quadrante da matriz SWOT. Afim de tornar mais explicativa a utilização de cada fator, foi inserida uma coluna “Comentários”, que justificam a utilização destes.

Após a soma dos valores ponderados por quadrante, chegou-se ao resultado apresentado na Tabela 6.

Tabela 6. Segregação de incidentes por programa

Forças	Ameaças	Fraquezas	Oportunidades
3,90	3,25	1,80	4,60

Fonte: Elaborado pelo autor.

E com os dados da Tabela 6, buscou-se representar os números em um gráfico de radar, apresentado na Figura 15 a seguir.

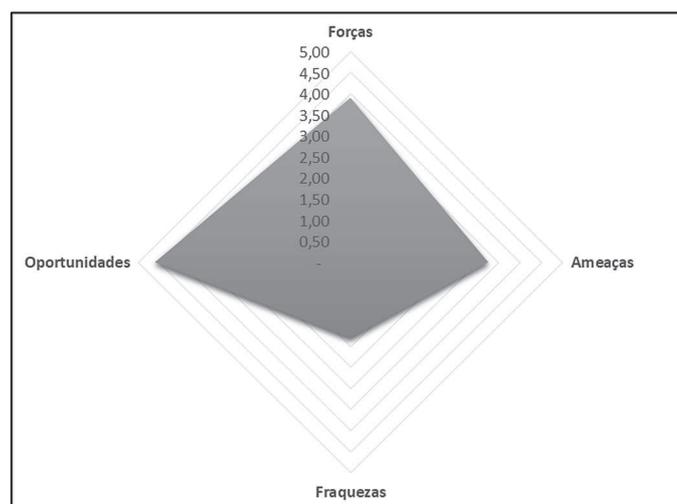


Figura 15 – Representação SWOT de fatores para elicitação de requisitos

Fonte: Elaborado pelo autor.

Como pode-se observar na Figura 15, os fatores apresentados no quadrante representando os pontos fortes na elicitação de requisitos da empresa XYZ, podem ser classificados como “Bons”, caminhando para a “Excelência”, entretanto, evidencia-se no quadrante das fraquezas que estes fatores precisam ser melhorados, de modo a somar nos pontos fortes. É importante ressaltar que o nível de oportunidades para melhorar o índice dos fatores é classificado como “Excelente”, contribuindo assim para o objetivo proposto neste estudo, ou seja, um aprimoramento na elicitação de requisitos. Não menos importante, as ameaças apresentadas possuem um valor bem representativo, o que estimula a empresa se envolver em questões públicas, visando incentivar a sociedade nos pontos que lhe convém, como por exemplo, mão-de-obra.

Com as informações de uma análise SWOT em mãos, pode-se vislumbrar possíveis estratégias afim de fortalecer o que a organização tem de melhor internamente e diminuir as fraquezas nas quais não esteja-se tão bem quanto desejado. Pensar em novas abordagens para o que enfraquece a organização, traz soluções diferenciadas e faz com que algumas ameaças sejam sanadas, movendo-se para a esquerda em referência à Figura 15.

5.5.2 Proposta de artefato para elicitação de requisitos

Conforme mencionado na subseção 5.2, nem toda demanda encaminhada ao setor de TI da empresa XYZ passa pela análise dos GTs de sistema, devido à fatores governamentais, econômicos, normativos, entre outros supracitados. Este fato pôde ser comprovado através da subseção 5.3, onde relaciona os dez programas com maior índice de incidentes, e destes, oito se referem aos fatores mencionados, o que reforça a necessidade de aprimoramento na etapa de elicitação de requisitos.

Entre os métodos ágeis apresentados nesta pesquisa, praticamente todos compartilham os seguintes aspectos com relação ao desenvolvimento ágil:

- Cliente incorporado à equipe, visando colaboração mútua;
- A comunicação é imprescindível e pode-se utilizar quaisquer meios que a facilitem;
- Liberdade e confiança na equipe para tomada de decisões;

- Segregação de funcionalidades, identificando as que geram maior retorno;
- Desenvolvimento de requisitos que agregam valor em primeiro lugar;
- Reuniões para discutir prioridades e acompanhar o andamento.

Dentre os aspectos mencionados acima, três se destacam nas fraquezas apresentadas pela análise SWOT da empresa XYZ, os quais correspondem à comunicação, segregação de funcionalidades e reuniões de prioridades e acompanhamento.

Quanto à comunicação, os pontos que evidenciam a necessidade de melhora, estão descritos como insipiência de usuários e carência de detalhes na elicitação de requisitos, pois dificultam a compreensão e entendimento das necessidades do sistema, gerando problemas de comunicação com o cliente.

A falta de documentação da regra de negócio e baixa flexibilidade para mudanças de requisitos inferem obstáculos para a segregação de funcionalidades, pois sem estas informações, não é possível identificá-las, tampouco averiguar o tipo de retorno esperado.

Sendo um dos aspectos mais importantes na aplicação de métodos ágeis, percebe-se que as reuniões para discussão de prioridades e acompanhamento não são realizadas com frequência, devido aos pontos de fraqueza “tempo para desenvolvimento” e “pouca antecipação de problemas”, pois é possível discutir estes pontos em breves reuniões.

Atualmente as demandas mencionadas acima chegam ao analista através do sistema mencionado na introdução, chamado “Service Desk”. É por este sistema que as demandas classificadas como urgentes e/ou exclusas da análise por GTs são enviadas pelos usuários/clientes.

A Figura 16 a seguir, apresenta a tela inicial de cadastro de uma nova solicitação de demanda, onde é necessário informar seu grupo de serviço, sistema, programa e assunto. Através destas informações, o programa já filtra demandas similares existentes, visando informar o usuário que a mesma demanda já pode ter sido enviada por outra pessoa.

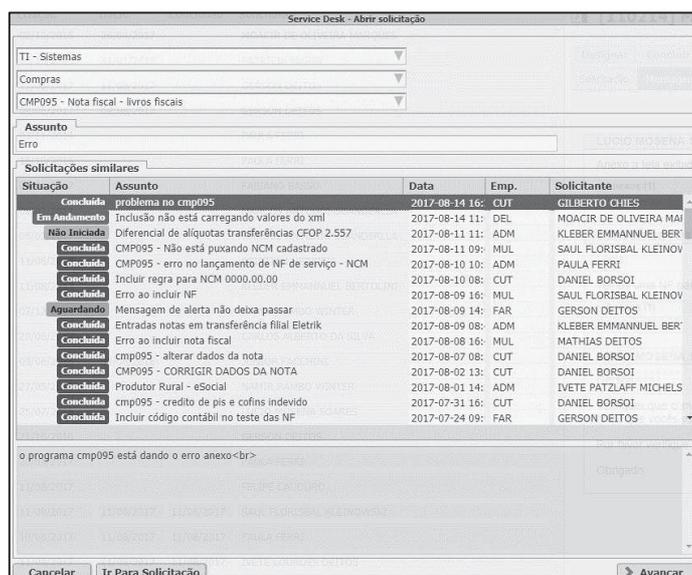


Figura 16 – Inclusão de solicitação de demanda

Fonte: Service Desk da empresa XYZ.

Após informar estes dados preliminares, o sistema avança para a área de descrição da demanda, representada pela Figura 17. Nesta etapa da solicitação de demanda, é necessário informar a empresa que está cadastrando a solicitação, o solicitante, o tipo de demanda, prioridade, data de planejamento, descrição e anexos.

Para descrever a solicitação de demanda e seus requisitos, é disponibilizado aos usuários uma área livre para descrição à seu modo, bem como para os anexos, pois não há modelos propostos para estes passos.

Figura 17 – Descrição de demandas e requisitos

Fonte: Service Desk da empresa XYZ.

Uma vez que a solicitação de demanda tenha sido inserida no sistema, é possível manter a comunicação com o solicitante através de mensagens (Figura 18), estas por sua vez, são enviadas a todos participantes da demanda, os quais se fazem presentes na opção de vínculos, também presente na Figura 18.

Figura 18 – Comunicação com envolvidos da demanda

Fonte: Service Desk da empresa XYZ.

Conforme visto nesta pesquisa, uma área livre para descrição da demanda e seus requisitos não tiveram o resultado esperado, pois há várias lacunas de detalhes a serem preenchidas antes do atendimento à demanda, visando evitar incidentes. Deste modo, foi proposto um artefato para elicitação de requisitos junto ao solicitante, antes de iniciar o atendimento e, que seja utilizado em conjunto com breves reuniões para definição de

prioridades e acompanhamento. Este artefato encontra-se no Apêndice II e foi validado frente aos *stakeholders* chave, selecionados pelo pesquisador. Na seção seguinte, será apresentado o *feedback* a respeito do artefato.

5.6 Feedback do artefato proposto

Conforme já mencionado nas seções anteriores, a intenção desta pesquisa é que o artefato para elicitação de requisitos venha a ser utilizado como entrada para o desenvolvimento de um assistente para elicitação no sistema "Service Desk". Diante deste objetivo, o artefato proposto foi apresentado aos *stakeholders* previamente selecionados, e a eles foi feita a seguinte pergunta: O artefato proposto para elicitação de requisitos fomenta a qualidade do *software* na empresa XYZ?

As respostas possíveis para a pergunta mencionada era: Sim, Indiferente, Não ou Ausente, para quando o *stakeholder* estivesse fora da empresa. A Tabela 7 a seguir, apresenta o resultado mapeado deste questionamento.

Tabela 7. Mapeamento do feedback dos stakeholders

	O artefato proposto para elicitação de requisitos fomenta a qualidade do software na empresa XYZ?			
<i>Stakeholder</i>	Sim	Indiferente	Não	Ausente
Stk01				X
Stk02	X			
Stk03	X			
Stk04	X			
Stk05			X	
Stk06	X			
Stk07	X			
Stk08	X			
Stk09		X		
Stk10	X			
TOTAL	7	1	1	1

Fonte: Elaborado pelo autor.

De acordo com o resultado, percebe-se que o artefato teve boa adesão, pois 70% com 10% de indiferença pode ser interpretado como 80% de adesão, sem contar que um dos *stakeholders* estava ausente durante a pesquisa.

6. Conclusão

A condução do mapeamento e análise da etapa de elicitação de requisitos frente aos métodos ágeis proporcionou uma considerável imersão no mundo ágil metodológico, mesmo o autor não utilizando ativamente os métodos ágeis no âmbito profissional. Além disso, foi possível realizar uma revisão nos conhecimentos com relação ao processo de engenharia de *software* e os aspectos influenciáveis na qualidade do desenvolvimento de *software*.

A partir do levantamento de informações acerca dos métodos ágeis é possível responder a pergunta norteadora de pesquisa - Como a elicitação de requisitos baseada em métodos ágeis pode aprimorar a qualidade do *software* na organização XYZ?

Identificou-se 3 aspectos para elicitação de requisitos em métodos ágeis que podem aprimorar a qualidade do *software* (apresentados na subseção 2.3), e destaca-se o seguinte:

- A geração de documentação é essencial para as organizações, pois contribuem para a análise e compreensão das regras de negócio, ao mesmo passo que, gerada sem gestão de contribuição, torna-se burocrática e pesada;
- Uma boa segregação de funcionalidades torna o desenvolvimento prático e ágil, facilita a identificação das prioridades de desenvolvimento e agrega muito valor ao negócio, sem contar na satisfação do cliente, o qual recebe o sistema desejado antes do esperado com suas funcionalidades principais;
- O segredo para o sucesso de um projeto depende de muitos aspectos intrínsecos ao processo de desenvolvimento, mas nem um outro é tão essencial quanto a comunicação, pois as informações geradas como entrada para o desenvolvimento de um sistema são obtidas através de pessoas, são elas que fazem o sucesso ou o fracasso de um projeto, por isso é primordial que uma organização mantenha seus colaboradores comprometidos, unidos e confiantes para atingir os objetivos determinados.

O estudo também permitiu ao pesquisador se aprofundar no processo de elicitação de requisitos da empresa XYZ e atender aos objetivos específicos (apresentados na seção 1).

Através das subseções 5.2 e 5.5 pôde-se identificar junto à empresa as técnicas e métodos utilizados na etapa de elicitação de requisitos, onde mapeou-se os pontos fortes e os pontos fracos através de uma análise SWOT. Buscando comprovação da premissa frente à ineficácia da etapa de elicitação dos requisitos, foi possível evidenciar e enfatizar a necessidade de aprimoramento através da subseção 5.3. Também foi proposto um artefato para auxiliar na condução de reuniões visando a elicitação de requisitos, conforme apresenta a subseção 5.5.2 e o Apêndice I. Baseado em métodos ágeis, propôs-se um artefato de certa forma simples, visando evitar um abruptamento na rotina atual de desenvolvimento da organização. A ideia principal é fazer com que o analista não perca o foco da reunião e que informações importantes não passem despercebidas. Este artefato foi avaliado junto aos gerentes do setor de tecnologia da informação (TI), e de acordo com o *feedback* apresentado na subseção 5.6, teve boa adesão à utilização.

O estudo permitiu gerar uma contribuição quanto ao conhecimento acerca da elicitação de requisitos na engenharia de *software*. Também pode ser utilizado para sanar algumas dúvidas que outras organizações possuem quanto à utilização de métodos ágeis, se irá contribuir ou não em seus processos.

Inicialmente, além da etapa de elicitação de requisitos, seria abordado o processo de gestão dos requisitos. Entretanto, devido à limitação de tempo para elaboração da pesquisa e do volume de informações que isso poderia acarretar, o estudo limitou-se à etapa de elicitação de requisitos.

Assim, como trabalhos futuros, seria muito interessante e proveitoso realizar um trabalho com relação à gestão dos requisitos, pois além de ser um tema interessante, irá

gerar conhecimento acerca deste processo que também influencia na elicitação, visto que faz parte do ciclo da engenharia de *software*.

Referências

- Agile Manifesto. (2001) Manifesto para Desenvolvimento Ágil de Software. Disponível em: <<http://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 03 ago. 2017.
- Batista, E. A. (2003) Uma Taxonomia Facetada para Técnicas de Elicitação de Requisitos. 2003. 164p. Dissertação (Mestrado em Computação na área de Engenharia de Computação) - Instituto de Computação, UNICAMP, Programa de Pós-Graduação em Informática.
- Brooks, F. P. (1987) "No Silver Bullet: Essence and Accidents of Software Engineering," *Computer*, Vol. 20, No. 4 pp. 10-19.
- Chiavegatti, N. C.; PETRI, G. (2014) EAReq-Game: Um Jogo Educacional para o Ensino de Elicitação e Análise de Requisitos. Anais do EATI - Encontro Anual de Tecnologia da Informação e Semana Acadêmica de Tecnologia da Informação, Frederico Westphalen-RS, Brasil, Novembro 10-14.
- Foggetti, C. (2014) Gestão ágil de projetos. São Paulo: Education do Brasil. (Coleção Bibliografia Universitária Pearson).
- Gil, A. C. (2002) Como elaborar projetos de pesquisa. São Paulo: Atlas.
- Gomes, A. S.; Wanderley, E. G. (2003) Elicitando requisitos em projetos de Software Educativo. In: IX Workshop de Informática na Escola - WIE, 2003, Pernambuco. Anais... Recife: WIE.
- Highsmith, J. A. (1999) Adaptive software development: a collaborative approach to managing complex systems / James A. Highsmith, III; foreword by Ken Orr.
- Hirama, K. (2011) Engenharia de software: qualidade e produtividade com tecnologia. 1.ed. Rio de Janeiro: Elsevier. 210p.
- Kaplan, B.; Duchon, D. (1988) Combining qualitative and quantitative methods in information systems research: a case study. *MIS Quarterly*, v. 12, n. 4.
- Koscianski, A.; Soares, M. dos S. (2007) Qualidade de Software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2.ed. São Paulo: Novatec. 395p.
- Lima, M. M. de. (2013) PREÁ - Um processo de desenvolvimento de software educacional baseado em metodologias ágeis. Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual do Paraíba, Centro de Ciências e Tecnologias.
- Machado, R.T.M. (2005) Estratégia e competitividade em organizações agroindustriais. Lavras: UFLA/FAEPE.
- Pressman, R. S. (2011) Engenharia de Software: uma abordagem profissional, 7.ed. – Dados Eletrônicos - Porto Alegre: AMGH.
- Silva, E. L. da. (2001) Metodologia da pesquisa e elaboração de dissertação/Edna Lúcia da Silva, Estera Muszkat Menezes. – 3. ed. rev. atual. – Florianópolis: Laboratório de Ensino a Distância da UFSC. 121p.

- Silva, R. C. da; Benitti, F. B. V. (2011) Padrões de Escrita de Requisitos: um mapeamento sistemático da literatura. Anais do WER11 - Workshop em Engenharia de Requisitos, Rio de Janeiro-RJ, Brasil, Abril 28-29, pp 259-272.
- Sommerville, I. (2011) Engenharia de Software. 9. ed. São Paulo. Pearson Prentice Hall.
- Souza, V. C. de. (2011) Engenharia de requisitos de software. São Leopoldo: Unisinos.
- Torres, R. R. (2000) Estudo sobre os Planos Amostrais das Dissertações e Teses em Administração da Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo e da Escola de Administração da Universidade Federal do Rio Grande do Sul: uma contribuição crítica. 2000. 246 p. Dissertação (Mestrado em Administração) – Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo.
- Vieira, S. R. C. (2012) REMO - Uma Técnica de Elicitação de Requisitos Orientada pela Modelagem de Processos de Negócios. 129p. Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas, Instituto de Computação, Programa de Pós-Graduação em Informática, 2012.
- Yin, R. K. (2001) Estudos de Caso: planejamento e métodos. Porto Alegre: Bookman.
- Young, R. R. (2002) Recommended Requirements Gathering Practices. CROSSTALK The Journal of Defense Software Engineering. Disponível em: <<http://staff.unak.is/andy/Year2%20Object%20Oriented%20Methods/HomeworkTests/RecReqGatPra.pdf>>. Acesso em: 02 abr. 2017.

APÊNDICE I

Tabela com atribuição de pesos aos fatores para elicitação de requisitos

Fatores	Peso	Classificação	Peso x Classificação	Comentários
Forças				
Comprometimento da equipe	20%	4	0,80	Todos demonstram comprometimento.
Auto gerenciamento das equipes, maior autonomia, disciplina e regularidade	10%	3	0,30	As equipes tomam a frente em busca de soluções.
Respostas rápidas às demandas	10%	5	0,50	Satisfação do cliente é o objetivo principal.
Foco e maximização do ROI (Retorno do Investimento) e do Valor de Negócio	20%	4	0,80	Quanto mais retorno, maior a importância.
Participação ativa dos usuários	10%	4	0,40	Mostram boa intenção de ajudar.
Habilidade técnica dos desenvolvedores	10%	4	0,40	Todos apresentam boas qualificações.
Condução de entrevistas	5%	3	0,15	A elicitação em presença do cliente tem fluência.
Produtividade	5%	4	0,20	Está na cultura da organização.
Transparência e visibilidade do status do projeto	5%	4	0,20	Todos podem acompanhar seus projetos.
Prototipação de sistemas	5%	3	0,15	Quando necessário é utilizado e reaproveitado no projeto.
TOTAL	100%	38	3,90	
Fraquezas				
Tempo para desenvolvimento	10%	2	0,20	Pode piorar a qualidade do software.
Insipiência de usuários	10%	2	0,20	Dificultam a comunicação.
Carência de detalhes na elicitação de requisitos	20%	2	0,40	Aumentam os riscos de incidentes.
Falta de documentação da regra de negócio	20%	2	0,40	Dificultam a compreensão para análise.
Pouca flexibilidade para mudanças de requisitos	10%	2	0,20	Pequenas mudanças afetam vários sistemas.
Entrega incremental apenas em módulos do sistema	10%	1	0,10	Os programas são desenvolvidos objetivando o atendimento em 100% dos requisitos.
Problemas de comunicação com o cliente	10%	2	0,20	Frequentemente há conflitos na comunicação.
Falta de inspeção e adaptação do processo em busca da melhoria contínua e a redução dos desperdícios	5%	1	0,05	Em meio à quantidade de demandas, não se investe tempo na melhoria do processo.
Pouca antecipação de problemas e postergação da tomada de ações	5%	1	0,05	Em muitas demandas não se avalia o impacto futuro.
TOTAL	100%	15	1,80	
Oportunidades				
Treinamentos	50%	5	2,50	Aprimoram as habilidades dos colaboradores.

Aquisição de software	20%	3	0,60	Auxilia no atendimento às demandas.
Novas tecnologias	30%	5	1,50	Aumentam a qualidade dos softwares.
TOTAL	100%	13	4,60	
Ameaças				
Desvalorização da moeda	25%	3	0,75	Com a desvalorização da moeda, empresas que exportam e importam produtos podem ter prejuízo.
Mudança na legislação	50%	3	1,50	Uma mudança na legislação do TI pode influenciar no tempo para execução dos projetos.
Escassez de mão-de-obra	25%	4	1,00	Sem mão-de-obra não há como atender as demandas.
TOTAL	100%	10	3,25	

APÊNDICE II

Artefato guia para elicitação de requisitos

Empresa XYZ	Elicitação de Requisitos	Solicitação N°:
Data da reunião:	dd/mm/yyyy	
Participantes:	
Demanda/Problema		
<i>Indicar o que está acontecendo ou descrever a solicitação de demanda.</i>		
Causas/Motivo		
<i>Indicar o que pode estar causando o comportamento, ou o motivo pelo qual é necessária a nova funcionalidade.</i>		
Importância/Retorno esperado		
<i>Especificar o quão importante se faz a funcionalidade/correção do incidente.</i>		
Plano/Solução/O que fazer		
<i>Indicar como deve ser atendida a demanda.</i>		
Protótipo/Modelo/Opções informadas		
<i>Indicar os modelos para implementação ou opções para simulação de incidentes.</i>		
Requisitos/Condições para a solução		
<i>Indicar os meios para chegar à solução, formas de cálculo, regras de negócio, entre outros.</i>		
Como aferir o resultado?		
<i>Indicar de que modo o resultado esperado pode ser averiguado.</i>		
Impacto (caso não seja implementado(a) - RISCO		
<i>Indicar os riscos que a organização corre se não desenvolver/corrigir a funcionalidade.</i>		
Observações		
<i>Indicar qualquer informação adicional referente ao projeto.</i>		