

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE ESPECIALIZAÇÃO EM GESTÃO DA TECNOLOGIA DA INFORMAÇÃO
E COMUNICAÇÃO

JONHATAN LUIZ DE SOUZA HAISI

**MODELO DE APLICAÇÃO DE INTELIGÊNCIA COLETIVA NO
LEVANTAMENTO DE REQUISITOS DE SOFTWARE**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA

2016

JONHATAN LUIZ DE SOUZA HAISI

**MODELO DE APLICAÇÃO DE INTELIGÊNCIA COLETIVA NO
LEVANTAMENTO DE REQUISITOS DE SOFTWARE**

Monografia do Curso de Especialização em Gestão da Tecnologia da Informação e Comunicação, do Departamento Acadêmico de Eletrônica (Daeln), da Universidade Tecnológica Federal do Paraná (UTFPR), como requisito parcial para obtenção do certificado de Especialista.

Orientador: Prof. Alexandre Graeml

CURITIBA

2016

JONHATAN LUIZ DE SOUZA HAISI

MODELO DE APLICAÇÃO DE INTELIGÊNCIA COLETIVA NO LEVANTAMENTO DE REQUISITOS DE SOFTWARE

Este trabalho de conclusão de curso foi apresentado no dia 10 de outubro de 2016, como requisito parcial para obtenção do certificado de Especialista em Gestão da Tecnologia da Informação e Comunicação, expedido pela Universidade Tecnológica Federal do Paraná. O estudante foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Msc. Alexandre J. Miziara
Coordenador de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Curitiba, 10 de outubro de 2016

Prof. Msc. Alexandre J. Miziara
UTFPR

Prof. Alexandre Graeml - UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

AGRADECIMENTOS

Primeiramente agradeço ao meu orientador Alexandre Graeml. Sou grato pelo seu esforço e dedicação ao me auxiliar neste trabalho. Sem dúvidas sua orientação foi determinante para conclusão desse estudo e do resultado final obtido. Minha amiga Karin Quandt também me ajudou muito, principalmente com revisões e sugestões. Sua ajuda foi muito significativa e fez grande diferença em meu trabalho. Minha irmã Aline Haisi literalmente esteve ao meu lado em cada linha escrita. Sua parceria sempre foi muito importante para mim e um incentivo a seguir cada vez mais longe na luta pelos meus objetivos.

Meu pai e meu avô possuem um papel muito importante na minha formação acadêmica e desenvolvimento pessoal. Eu sempre os tive como exemplo, tanto na dedicação aos estudos quanto na busca por seus objetivos. Os dois me incentivaram muito a me dedicar aos estudos e tiveram papel fundamental na minha carreira e na minha vida.

Agradeço também à minha família e amigos pelo apoio durante o decorrer do curso. Sem dúvidas atingir meus objetivos seria muito mais difícil sem a força que recebi durante todo o curso e execução desse trabalho.

“Muitas das coisas mais importantes do mundo foram conseguidas por pessoas que continuaram tentando quando parecia não haver mais nenhuma esperança de sucesso”

Dale Carnegie (pensador)

RESUMO

HASI, Jonhatan Luiz de Souza. Modelo de aplicação de inteligência coletiva no levantamento de requisitos de software: 2016. 57 f. Monografia (Curso de Especialização em Gestão da Tecnologia da Informação e Comunicação), Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Esse estudo buscou uma forma de adaptar a tarefa de levantamento de requisitos de software para que possa ser usada uma abordagem orientada à inteligência coletiva. Para tanto foi feito um estudo nas áreas de engenharia de requisitos e inteligência coletiva buscando encontrar pontos em comum para o desenvolvimento de um modelo de processo de engenharia de requisitos. Para o desenvolvimento do processo para levantamento de requisitos foram estudadas, principalmente, as tarefas da engenharia de requisitos definidas por Pressman, o conceito de inteligência estratégica antecipativa coletiva e a ferramenta *puzzle* para criação de sentido coletivo. As tarefas da engenharia de requisitos, a inteligência estratégica antecipativa coletiva e a ferramenta *puzzle* foram fundidas em um único processo com o objetivo de propor uma nova forma de coletar informações sobre os serviços que o software deve oferecer, a partir de trabalho colaborativo de todos que podem auxiliar no levantamento de informações. Buscou-se projetar um processo otimizado do ponto de vista da aderência dos requisitos de software às necessidades organizacionais e suas atividades operacionais.

Palavras-chaves: inteligência coletiva, inteligência antecipativa, engenharia de requisitos de software.

ABSTRACT

This study aims at adapting the software requirements specification tasks so that they can benefit from the use of a collective intelligence approach. In order to do that, both the area of requirements specifications and the area of collective intelligence were reviewed to find common groups and/or intersections on which to base the development of a new process model for the specification tasks relying on the participation of many Pressman's software requirement analysis tasks deserved special attention, as well as the collective anticipative intelligence concept and the *puzzle* tool, used to generate a collective sense. Pressman's tasks for software requirement analysis, collective anticipative strategic intelligence and the *puzzle* tool were all merged in a single process with the intent of obtaining a new way of collecting information about the service that a software will have to perform, based on the work of all those who can help raising information. An attempt was made to ensure capturing software requirements that are aligned to the organizational needs and the operational activities that lead to their accomplishment in an optimized way.

Keywords: collective intelligence, anticipative intelligence, requirements analysis, software development.

LISTA DE ILUSTRAÇÕES

Figura 1: Diagrama de classes de um pacote que possui um conjunto de Casos de Uso, Atores, e um subsistema.....	29
Figura 2: Diagrama de atividade que faz a sequência de ações fácil de entender....	30
Figura 3:Matriz representando os atores e os temas em que irão atuar.	32
Figura 4: <i>Framework Puzzle</i>	35
Figura 5: Exemplo de <i>Puzzle</i>	36
Figura 6: Representação do ciclo de levantamento de requisitos com inteligência coletiva.	44
Figura 7: Mapa mental exemplificando a segregação da base de conhecimento de um sistema ERP em temas, categorias e subcategorias.	51
Figura 8:Exemplo de <i>puzzle</i> para o tema compras de um ERP.	54

LISTA DE QUADROS

Quadro 1: Exemplo de estrutura de um documento de requisitos.....	28
Quadro 2: Matriz RACI da etapa de levantamento.....	50
Quadro 3: Matriz RACI da etapa de elaboração.....	54
Quadro 4: Matriz RACI da etapa de especificação.....	56
Quadro 5: Matriz RACI da etapa de validação	59
Quadro 6: Matriz RACI do ciclo contínuo	63

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Contextualização	11
1.2	Tema	12
1.3	Problema	13
1.4	Objetivos	14
1.4.1	Objetivo geral	14
1.4.2	Objetivos específicos	14
1.5	Escopo e delimitação do trabalho	15
1.6	Justificativa	16
1.7	Procedimentos Metodológicos	16
1.8	Embasamento Teórico	17
1.9	Estrutura do Trabalho	17
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Engenharia de requisitos	19
2.1.1	Conceitos e objetivos	19
2.1.2	Tarefas da engenharia de requisitos	20
2.1.3	<i>Stakeholders</i> de um projeto de software	23
2.1.4	Processos de engenharia de requisitos	23
2.1.5	Documentação de requisitos	27
2.2	Inteligência coletiva	31
2.2.1	Conceitos e objetivos	31
2.2.2	Inteligência estratégica antecipativa coletiva	32
2.2.3	A ferramenta <i>Puzzle</i>	33
3	APRESENTAÇÃO E ANÁLISE DE RESULTADOS	38
3.1	Visão geral do modelo proposto	38
3.2	Papéis	38
3.2.1	Colaborador	39
3.2.2	Animador	40
3.2.3	Negociador	41
3.2.4	Analistas de requisitos	42
3.3	O ciclo do processo	43
3.3.1	Concepção	44
3.3.2	Levantamento	46
3.3.3	Elaboração	50
3.3.4	Especificação	54
3.3.5	Validação	57
3.3.6	Ciclo de atuação contínua	59
3.4	Artefatos	64
3.4.1	Objetivo do projeto	64
3.4.2	Base de conhecimento	65
3.4.3	Temas	65
3.4.4	<i>Puzzle</i>	66
3.4.5	Documento de requisitos	67
3.4.6	Lista de aceitação dos requisitos	67

4 CONSIDERAÇÕES FINAIS.....	68
REFERÊNCIAS.....	70

1 INTRODUÇÃO

1.1 Contextualização

A tecnologia da informação vem evoluindo com o passar dos anos e tendo como um de seus objetivos prover ferramentas para criar diferencial de mercado para empresas que utilizam seus serviços.

Para Carvalho (2010) a engenharia de requisitos surgiu para minimizar problemas de projetos de software ocasionados por requisitos de software. Franceto (2005) afirma que, no processo de desenvolvimento de software, definir requisitos que sejam compreensíveis por todas as partes envolvidas é um problema de difícil solução. Com isso entende-se a engenharia de requisitos como algo de grande importância para projetos de software, mas que não se trata de uma atividade de fácil resolução.

Todo processo de levantamento de requisitos deve seguir os princípios da engenharia de software usando adequadamente os métodos, técnicas e ferramentas disponíveis para essa tarefa (FRANCETO, 2005). Muitas técnicas e abordagens para levantamentos de requisitos podem ser encontradas na literatura.

Pressman (2011) expõe a existência de várias abordagens para coleta de requisitos de software. Porém, aponta que em geral essas seguem algumas diretrizes básicas, tais como: reuniões com participação de engenheiros e interessados, que devem seguir regras preestabelecidas e para as quais é sugerida uma agenda formal para cobrir os pontos importantes. A partir da análise de Pressman percebe-se a inviabilidade, principalmente em empresas de grande porte e com grande quantidade de funcionários, de envolver todos os funcionários da organização nas reuniões de levantamento de requisitos, limitando as definições aos conhecimentos dos presentes nessas reuniões.

1.2 Tema

Um entre os vários processos presentes em um projeto de desenvolvimento de software é o levantamento de requisitos. Segundo Pressman (2011, p. 126), “entender requisitos de um problema está entre as tarefas mais difíceis enfrentadas por um engenheiro de software”. “A engenharia de requisitos foi criada para definir todas as atividades envolvidas em descobrir, documentar e manter um conjunto de requisitos para um projeto de sistemas de software” (SOMMERVILLE e SAWYER, 1997, *apud* FRANCETO, 2005, p. 22).

Apesar dos vários métodos de levantamento de requisitos atuais, a falha no entendimento das necessidades do cliente ainda é um motivo de insucesso de projetos de desenvolvimento de sistemas de informações (PINTO, 2002).

Certamente parece bem simples - pergunte ao cliente, aos usuários e aos demais interessados quais são os objetivos para o sistema ou produto, o que deve ser alcançado, como o sistema atende as necessidades da empresa e, por fim, como o sistema deve ser utilizado no dia a dia, mas isso não é simples (PRESSMAN, 2011, p. 128).

Uma série de problemas pode ser identificado no levantamento de requisitos: problemas de escopo, entendimento e volatilidade (Christel e Kang, 1992, *apud* PRESSMAN, 2011, p. 128). Diante disso pode-se dizer que um dos motivos do não atendimento das necessidades de negócio pelo software é a dificuldade enfrentada por analistas de tecnologia da informação em identificar todos os requisitos a serem atendidos, principalmente quando o conhecimento e artefatos se encontram difusos em uma organização. Com isto percebe-se a inviabilidade para a equipe de analistas de sistemas atuar em todas as fontes de informações.

A partir do contexto apresentado, esse trabalho tem como tema genérico a engenharia de requisitos, porém refinando-o, buscando verificar a possibilidade da utilização do trabalho colaborativo para documentação de processos organizacionais e levantamento de requisitos de software. O tema refinado desta pesquisa é engenharia de requisitos utilizando inteligência coletiva.

1.3 Problema

Apesar de todos os benefícios disponibilizados pelas novas tecnologias, esses nem sempre são percebidos. “Em muitos casos, a TI é vista como um ‘mal necessário’: concorda-se em tê-la, mas busca-se minimizar cada vez mais o transtorno de aumentar o seu total de recursos investidos” (LEITE, 2004, *apud* LUNARDI e DOLCI, 2006, p. 1). Alguns fatores contribuem para esta visão: um deles é a grande taxa de insucesso de projetos de TI. Segundo Pinto (2002, p. 19), “somente 10% dos projetos de software são entregues com sucesso, dentro do orçamento e prazo estimados”. Esse mesmo autor aponta um dos motivos para insucessos de projetos de software (PINTO, 2002, p. 60):

O gerenciamento de requisitos foi criado para simplificar o desenvolvimento de produtos, para reduzir seu custo e também reduzir o risco intrínseco associado com o desenvolvimento de sistemas de informação. Em vez disso, o gerenciamento de requisitos tem se tornado cada vez mais uma atividade propensa a erros. O gerenciamento de requisitos é geralmente considerado uma das maiores causas de falhas de produtos.

Sobre os problemas de entendimento dos requisitos de software Pressman (2011, p. 128) afirma que:

Os clientes/usuários não estão completamente certos do que é preciso, têm um conhecimento inadequado das capacidades e limitações de seus ambientes computacionais, não possuem um entendimento completo do domínio do problema, têm problemas para transmitir suas necessidades ao engenheiro de sistemas, omitem informações que acreditam ser “óbvias”, especificam requisitos que conflitam com as necessidades de outros clientes/usuários ou especificam requisitos que são ambíguos ou impossíveis de ser testados.

A partir da afirmação de Pressman pode-se identificar barreiras no processo de levantamento de requisitos, entre elas: informação descentralizada e não documentada, processos empresariais desconexos, retenção de conhecimento por funcionários e não integração entre setores. Ainda sobre o levantamento de problemas de entendimento dos requisitos realizado por Pressman, conclui-se que: a construção de um sistema deve aproveitar ao máximo as habilidades e conhecimentos de todos os funcionários da empresa a que esse se destina, para assim maximizar a eficácia e eficiência do software produzido no atendimento das necessidades do negócio.

A partir da análise do problema apresentado por esse trabalho, verifica-se que, para aproveitar melhor os conhecimentos dos funcionários da empresa no processo de levantamento de requisitos de software, se tem a necessidade da apropriação do conhecimento e trabalho colaborativo de todos os envolvidos no projeto de desenvolvimento de software. Percebe-se, nisso, uma familiaridade com a proposta da inteligência coletiva, que compreende a busca por produzir algo a partir de esforços de um grupo de pessoas como uma de suas aplicações (LÉVY, 1994, *apud* PEREIRA, 2012, p. 5).

Diante disso, esse trabalho propõe a criar um modelo de aplicação de inteligência coletiva no levantamento de requisitos de software, baseado em estudos de aplicação da inteligência coletiva e engenharia de requisitos em contextos empresariais.

1.4 Objetivos

1.4.1 Objetivo geral

Este trabalho tem como objetivo propor um modelo de aplicação de inteligência coletiva para realização da engenharia de requisitos em projetos de software.

1.4.2 Objetivos específicos

A partir do estudo bibliográfico sobre inteligência coletiva e levantamento de requisitos de software, tem-se como objetivos específicos:

- Entender como funciona a engenharia de requisitos e o que é necessário para um levantamento de requisitos de software eficaz.

- Estudar técnicas, diretrizes e abordagens utilizadas na engenharia de requisitos e formas de aplicá-las.
- Identificar, a partir de estudo bibliográfico, características de sucesso na aplicação de inteligência coletiva.
- Propor, a partir das informações obtidas, um modelo de aplicação de inteligência coletiva à engenharia de requisitos de software.

1.5 Escopo e delimitação do trabalho

Esse trabalho visa a estudar a possibilidade da utilização da inteligência coletiva em ambientes empresariais para o levantamento de requisitos de software. Constatando-se a viabilidade, também busca-se apresentar uma proposta de aplicação de inteligência coletiva para levantamento de requisitos de software.

Para esse estudo, serão feitas pesquisas sobre casos de sucessos na utilização de inteligência coletiva em ambiente empresarial, estudo de técnicas de levantamento de requisitos de software já utilizadas por equipes de desenvolvimento de sistemas de informação e formas como pessoas não relacionadas à TI podem auxiliar no levantamento de requisitos de software.

Esse trabalho não visa a discutir todas as características de levantamentos de requisitos de software ou inteligência coletiva, limitando-se às minimamente necessárias para que se possa utilizar inteligência coletiva no suporte ao levantamento de requisitos. O resultado desse trabalho deve ser um modelo para a aplicação de inteligência coletiva na engenharia de requisitos de software. Porém esse modelo não será testado por essa pesquisa. O teste do modelo aqui proposto fica a cargo de trabalho futuro.

1.6 Justificativa

Apesar da grande dependência que empresas de todos os ramos têm da TI, o emprego da TI nem sempre consegue atender às necessidades do negócio (PINTO, 2002). A dificuldade enfrentada por equipes de TI no momento de extrair requisitos de software é um dos motivos de insucesso de projetos de desenvolvimento de sistemas de informação (PINTO, 2002).

O trabalho colaborativo na engenharia de requisitos de projetos de software pode vir a ser uma forma de mitigar problemas relacionados ao entendimento dos requisitos. Esses problemas derivados de falhas de entendimento, como os apontados por Pressman (2011) podem ter, na inteligência coletiva, uma nova abordagem de trabalho com benefícios para o projeto de software.

1.7 Procedimentos Metodológicos

Esse trabalho propõe um estudo sobre a aplicabilidade da inteligência coletiva na engenharia de requisitos de software, buscando descobrir a viabilidade da utilização dessa abordagem. O estudo terá como base técnicas de levantamento de requisitos, já aplicadas atualmente em projetos de software, e um estudo bibliográfico sobre casos de aplicação da inteligência coletiva.

Do ponto de vista da abordagem do problema o trabalho pode ser classificado como qualitativo, pois, de acordo com DALFOVO, LANA e SILVEIRA (2008, p. 9), “pesquisa qualitativa é aquela que trabalha predominantemente com dados qualitativos, isto é, a informação coletada pelo pesquisador não é expressa em números, ou então os números e as conclusões neles baseadas representam um papel menor na análise”.

Este trabalho envolve a análise de estudos de aplicação de inteligência coletiva e como o trabalho colaborativo pode contribuir na engenharia de requisitos. A partir do estudo bibliográfico busca-se identificar formas para que pessoas não relacionadas à tecnologia da informação possam realizar a engenharia de requisitos

de um projeto de software. Para que esse objetivo seja atingido os seguintes passos para elaboração do estudo foram definidos:

- Estudo bibliográfico sobre engenharia de requisitos.
- Estudo bibliográfico sobre abordagens de levantamento de requisitos.
- Estudo bibliográfico sobre inteligência coletiva.
- Estudo bibliográfico sobre aplicação da inteligência coletiva em ambiente corporativo.
- Estudo bibliográfico de práticas e abordagens complementares à engenharia de requisitos com foco em trabalho colaborativo.
- Desenvolvimento de um modelo de aplicação de inteligência coletiva em engenharia de requisitos usando como base as informações obtidas.

1.8 Embasamento Teórico

Relacionado a engenharia de requisitos foram utilizados como referencial os autores Pressman (2011), Carvalho (2010), Franceto (2005), Sommerville (2011), Pinto (2002), PMI (2013). Sobre inteligência coletiva foram usados como referencial teórico autores como Pereira (2012), Blanck (2014), Janissek-Muniz (2014), Bembem (2013), Santos (2013), Lozzi (1996), Xiberras (2010), Ferreira (2013), Rouibah (2002) e Ouli-Ali (2002).

1.9 Estrutura do Trabalho

O presente estudo terá a seguinte estrutura abaixo apresentada:

Capítulo 1 - Introdução: Será apresentado o tema, as delimitações da pesquisa, o problema e a premissa, os objetivos da pesquisa, a justificativa, os procedimentos metodológicos, as indicações para o embasamento teórico, e a estrutura geral do trabalho.

Capítulo 2 – Fundamentação Teórica: Será realizada uma pesquisa sobre inteligência coletiva aplicada às organizações, levantamento de requisitos e documentação de software.

Capítulo 3 – Apresentação e Análise dos Resultados: tendo como base os procedimentos metodológicos, neste capítulo serão descritos os resultados obtidos e feitas as devidas análises relacionados à aplicação de trabalho colaborativo para documentação de requisitos de software.

Capítulo 4 – Considerações finais: será retomada a pergunta de pesquisa e os seus objetivos e apontado como foram solucionados, respondidos, atingidos, por meio do trabalho realizado.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Engenharia de requisitos

2.1.1 Conceitos e objetivos

Requisitos de software são como são chamadas as descrições do que se espera de um sistema de informação. Eles são compostos pelos serviços que o software deve prover, além das restrições de seu funcionamento (SOMMERVILLE, 2011). Segundo o PMI (2013, p. 112):

Os requisitos incluem condições ou capacidades que devem ser atendidas pelo projeto ou estar presentes no produto, serviço ou resultado para cumprir um acordo ou outra especificação formalmente imposta. Os requisitos incluem as necessidades quantificadas e documentadas e as expectativas do patrocinador, cliente e outras partes interessadas.

Com o objetivo de levar a todas as partes interessadas um entendimento formal do problema a engenharia de requisitos, parte integrante da engenharia de software, conta com um conjunto de tarefas e técnicas usadas no levantamento e documentação de requisitos de software (PRESSMAN, 2011). Thayer e Dorfman (1997, *apud* SOMMERVILLE, 2011, p.127) afirmam que:

A engenharia de requisitos fornece o mecanismo apropriado para entender aquilo que o cliente deseja, analisando as necessidades, avaliando a viabilidade, negociando uma solução razoável, especificando a solução sem ambiguidades, validando a especificação e gerenciando as necessidades à medida que são transformadas em um sistema operacional.

O objetivo da engenharia de requisitos, segundo Pressman (2011), é auxiliar todos os envolvidos a terem um melhor entendimento do problema a ser resolvido, documentando e formalizando os requisitos identificados em uma série de artefatos. Sommerville (2011) apresenta alguns artefatos de engenharia de requisitos como, por exemplo, histórias de usuários, casos de uso, documento de requisitos e descrição textual.

A partir das análises feitas por Sommerville, Pressman, Thayer e Dorfman pode-se definir a engenharia de requisitos como: uma subárea da engenharia de

software que tem como objetivo entender os requisitos de negócio, documentando-os em um formato que possa ser compreendido por todas as partes interessadas.

2.1.2 Tarefas da engenharia de requisitos

Segundo Pressman (2011) a engenharia de requisitos pode ser dividida em sete tarefas distintas que podem ser executadas simultaneamente. São elas:

- Concepção

A primeira tarefa da engenharia de requisitos definida por Pressman (2011) trata dos motivos que dão início a um projeto de software. Ele limita essa tarefa ao evento que dá origem à ideia de um novo projeto de desenvolvimento de software. Quanto ao evento de concepção do projeto de software, Pressman (2011, p. 127) explica:

Em alguns casos, uma conversa informal é tudo o que é preciso para precipitar um trabalho de engenharia de software. Porém, em geral, a maioria dos projetos começa quando é identificada uma necessidade do negócio ou descoberto um serviço ou mercado potencial.

- Levantamento

Essa tarefa busca levantar as necessidades do cliente. Os problemas que devem ser resolvidos e objetivos a serem alcançados são identificados como requisitos a serem atendidos. Os requisitos podem ser divididos em dois níveis, sendo eles: requisitos de usuário e requisitos de sistema (SOMMERVILLE, 2011). Apesar de possuírem o mesmo significado, Pressman (2011) denomina os níveis de requisitos de forma diferenciada: requisitos de alto nível, correspondendo aos requisitos de usuário, e requisitos de baixo nível, correspondendo aos requisitos de sistema.

Segundo Pressman (2011) o levantamento de requisitos de usuário é o foco dessa tarefa. Esses requisitos são as necessidades macro do usuário representadas de forma escrita ou por diagramas (SOMMERVILLE, 2011).

A partir das análises de Pressman (2011) e Sommerville (2011) percebe-se requisitos de usuários como abstrações em alto nível das funcionalidades que o sistema deve possuir, sem se focar em seus detalhes. Eles servem a pessoas que não estão interessadas em detalhes de implementação e funcionamento. São uma visão geral das funcionalidades que o sistema deve oferecer e são mais focadas em o que deve ser oferecido do que em como funcionará.

- **Elaboração**

Segundo Pressman (2011), essa tarefa tem como objetivo a exploração e refinamento de todas as informações coletadas até que sejam gerados os requisitos de baixo nível. Sommerville (2011) denomina esses requisitos como requisitos de sistema. Nessa tarefa é elaborado o documento de requisitos que deve conter as definições exatas dos requisitos a serem implementados, pois esse documento é uma espécie de contrato entre o cliente e os desenvolvedores (SOMMERVILLE, 2011).

Segundo o PMI (2013, p.117):

Os requisitos podem começar em um alto nível e tornarem-se progressivamente mais detalhados conforme mais informações sobre estes são conhecidas. Antes das linhas base serem estabelecidas, os requisitos devem ser não ambíguos (mensuráveis e passíveis de testes), rastreáveis, completos, consistentes e aceitáveis para as principais partes interessadas. O formato de um documento de requisitos pode variar de uma simples lista categorizada por partes interessadas e prioridades a formas mais elaboradas contendo um resumo executivo, descrições detalhadas e anexos.

Conforme Sommerville (2011) os requisitos de sistema podem ser divididos em requisitos funcionais ou não funcionais conforme sua relação com negócio:

Requisitos funcionais: São os serviços que o sistema deve prover, como o sistema vai reagir à entrada de informações e como o sistema deve se comportar em determinadas situações. Em alguns casos os requisitos devem descrever estados inválidos do sistema.

Requisitos não funcionais: São as restrições de serviços e funcionalidades disponibilizadas pelo sistema. Podem incluir: restrições de *performance*, de

processos de desenvolvimento e limitações imposta por padrões. Diferentemente dos requisitos funcionais, os não funcionais são aplicados ao sistema como um todo e não a serviços ou funcionalidades específicas.

- Negociação

Pressman (2011) define essa tarefa como sendo aquela em que os requisitos levantados são negociados quanto ao escopo e viabilidade de implementação, possibilitando que conflitos sejam resolvidos.

- Especificação

A especificação de software é a tarefa que visa a documentar, de forma escrita, os requisitos de sistema (PRESSMAN, 2011). O PMI (2013) descreve o objetivo da documentação de software como uma descrição de como os requisitos individuais atendem às necessidades de negócio.

Vários formatos de documentação podem ser empregados na especificação de requisitos. Pressman (2011, p.129) aponta como formatos de documentação válidos: “um documento por escrito, um conjunto de modelos gráficos, um modelo matemático formal, um conjunto de cenários de uso, um protótipo ou qualquer combinação dos fatores citados”.

- Validação

Segundo Pressman (2011) essa tarefa trata como os artefatos produzidos na engenharia de requisitos são validados quanto à qualidade. Inconsistência, ambiguidades, omissões ou quaisquer outros erros encontrados devem ser corrigidos para garantir que os requisitos estejam de acordo com a expectativa do cliente (PRESSMAN, 2011).

Sommerville (2011) considera a validação de requisitos como um processo que confere se os requisitos realmente atendem às necessidades do cliente. “A validação de requisitos é importante, pois erros em um documento de requisitos podem conduzir a um custo extensivo por retrabalho quando esses problemas são

descobertos durante o desenvolvimento ou depois que o sistema está em produção” (SOMMERVILLE, 2011, p. 110, tradução nossa).

- Gestão dos requisitos

O gerenciamento de requisitos é composto pelos processos que serão usados para gerenciar as tarefas de levantamento de requisitos com o objetivo de definir e documentar as necessidades das partes interessadas (PMI, 2013).

2.1.3 *Stakeholders* de um projeto de software

O PMI (2013) define *stakeholder* como um indivíduo, grupo ou organização que possui algum interesse em um projeto. Tal interesse pode ocorrer caso um indivíduo afete ou sinta-se afetado pelo projeto. Em um projeto deve ser feita a análise de parte interessadas, identificando os interesses relevantes ao projeto (PMI, 2013).

Pressman (2011) identifica a importância dos *stakeholders* para um projeto de software por esses terem a capacidade de contribuir com informações relevantes para o levantamento de requisitos. A existência de uma grande quantidade de interessados faz com que os requisitos passem por vários pontos de vista, o que pode ser benéfico para enriquecer os requisitos identificados.

2.1.4 Processos de engenharia de requisitos

2.1.4.1 Abordagens e técnicas de coleta de requisitos

Segundo o PMI (2013, p. 110), “coletar requisitos é o processo de determinar, documentar e gerenciar requisitos das partes interessadas, a fim de atender aos objetivos do projeto”. Existem muitos métodos de levantamento de requisitos disponíveis na literatura. Este trabalho se baseia nas seguintes técnicas e abordagens extraídas de publicações do PMI (2013), de Pressman (2011), Sommerville (2011) e OMG (2015):

- Entrevistas

Entrevista é uma forma técnica de levantamento de requisitos que ocorre a partir de conversas, formais ou não, em que o entrevistador realiza perguntas a um ou mais entrevistados e registra suas respostas (PMI, 2013). Essa técnica busca, a partir das entrevistas aos *stakeholders*, auxiliar na identificação de requisitos desejados. (PMI, 2013).

- Grupo de discussão

Grupo de discussão é uma reunião de engenheiros de requisitos e pessoas selecionadas com o objetivo de conversarem sobre os objetivos e expectativas sobre o produto (PMI, 2013).

- Oficinas facilitadas

Com foco na definição de requisitos multifuncionais e conciliação de diferenças de forma rápida, as oficinas facilitadas ocorrem com a reunião das partes interessadas chave, focadas na definição dos requisitos e discussões sobre as necessidades que o sistema deve atender (PMI,2013).

- Técnicas de criatividade em grupo

O PMI (2013) cita algumas técnicas de criatividade em grupo:

Brainstorming: é uma técnica usada para levantar ideias relacionadas aos requisitos do projeto e produto.

Técnica de grupo nominal: essa técnica estende o *brainstorming* adicionando a essa um processo de votação.

Mapas mentais: as ideias criadas por sessões de *brainstorming* são consolidadas em um mapa mental.

Diagramas de afinidade: agrupamento de um grande volume de ideias para que essas possam ser revisadas e analisadas.

Análise de decisão envolvendo critérios múltiplos: uma abordagem analítica sobre uma matriz de decisão visando avaliar e classificar as ideias.

- Técnicas de tomada de decisão em grupo

Algumas técnicas de tomada de decisão em grupo são citadas pelo PMI (2013). São elas:

Unanimidade: quando todos concordam com uma única decisão.

Maioria: quando a maior parte dos envolvidos no processo de decisão concordam com uma mesma opção.

Pluralidade: quando uma ideia possui a maior quantidade de adeptos, mesmo que não possua mais de 50% dos votos totais. Essa técnica é mais usada quando as possíveis escolhas ultrapassam duas opções.

Ditadura: quando o poder de decisão se concentra nas mãos de uma única pessoa.

- Questionário e pesquisas

O PMI (2013, p. 116) descreve questionários e pesquisas como:

Conjuntos de perguntas escritas, projetadas para acumular rapidamente informações de um grande número de respondentes. Os questionários e/ou pesquisas são mais apropriados para audiências variadas, quando uma resposta rápida é necessária, quando os respondentes estão geograficamente espalhados.

- Observações

Trata-se de observar as pessoas buscando analisar como elas desempenham suas atividades. Normalmente é realizado por um observador externo acompanhando um especialista (PMI, 2013).

- Protótipos

Um protótipo é um modelo do sistema construído para uma demonstração prévia do resultado da implementação de um ou mais requisitos (PMI, 2013). É um método para validação e refinamento de requisitos de software (PMI, 2013). O PMI (2013, p. 116) também afirma que: “já que o protótipo é tangível, ele permite que as partes interessadas façam experiências com um modelo do seu produto final ao invés de somente discutirem representações abstratas dos seus requisitos”.

- *Benchmarking*

Trata-se da descoberta de requisitos a partir da comparação entre práticas organizacionais com as de outras organizações semelhantes. Essa comparação também estimula a criação de ideias para melhorias internas e descoberta de melhores práticas (PMI, 2013).

- Análise de documentos

Trata-se de estudar e analisar os documentos produzidos e utilizados pelo cliente, buscando, nesses documentos, requisitos de software ou oportunidades de melhoria (PMI, 2013).

- Casos de uso

Casos de uso, ou cenários de uso, são uma espécie de roteiro de utilização do sistema, que descreve como este será utilizado por um grupo de usuários (PRESSMAN, 2011).

- Etnografia

É uma técnica observacional em que um analista passa por uma imersão no ambiente de trabalho em que o sistema será usado (SOMMERVILLE, 2011).

2.1.5 Documentação de requisitos

A documentação de requisitos tem como objetivo descrever como os requisitos vão atender às necessidades do cliente (PMI, 2013). Não existe um formato de documento de requisitos que seja universal, podendo esse ser, por exemplo, em formato de diagramas, documentos elaborados e até lista categorizada (PMI, 2013).

Muitas organizações desenvolvem seus próprios padrões de documentação de requisitos (SOMMERVILLE, 2011). As organizações podem projetar a forma como irão documentar os requisitos conforme suas necessidades específicas. Apenas devem atentar para que o propósito do documento de requisitos é oferecer uma descrição formal do que os desenvolvedores devem implementar (SOMMERVILLE, 2011).

Capítulo	Descrição
Prefácio	Deve definir o público alvo do documento e descrever seu histórico de versões, incluindo uma base lógica para a criação de uma nova versão e um sumário para as alterações feitas em cada versão
Introdução	Deve descrever a necessidade para o sistema. Deve descrever brevemente as funções do sistema e explicar como vai interagir com outros sistemas. Também deve descrever como o sistema se encaixa no contexto de negócio ou objetivos estratégicos da organização que contrata o sistema.
Glossário	Deve definir os termos técnicos usados no documento. Não se deve fazer premissas sobre a experiência e conhecimento do leitor.
Definições de requisitos de usuário	Aqui são descritos os serviços providos ao usuário. Os requisitos não funcionais também devem ser descritos nessa seção. Essa descrição pode usar linguagem natural, diagramas, ou outras notações que

	sejam compreensíveis pelo cliente. Padrões de produtos e processos que devem ser seguidos devem ser especificados.
Arquitetura do sistema	Este capítulo deve apresentar uma prévia de alto nível da arquitetura do sistema, apresentando a distribuição das funções através dos módulos do sistema. Componentes arquiteturais que são reutilizados devem ser destacados.
Especificação de requisitos de sistema	Deve descrever os requisitos funcionais e não funcionais com mais detalhes. Se necessário, detalhes adicionais podem ser adicionados aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidos.
Modelos de sistema	Pode incluir modelos gráficos do sistema demonstrando o relacionamento entre os componentes do sistema, o sistema, e seu ambiente. Exemplos de possíveis modelos são modelos de objetos, modelos de fluxo de dados, ou modelos semânticos de dados.
Evolução do sistema	Deve descrever as premissas fundamentais nas quais o sistema é baseado, e quaisquer mudanças previstas devido à evolução do hardware, mudanças nas necessidades dos usuários, e assim por diante. Essa seção é útil para designers de sistemas, pois isso pode ajudá-los a evitar decisões de <i>design</i> que devem restringir futuras alterações desejadas no sistema.
Apêndices	Devem prover detalhes, informações específicas que são relacionadas à aplicação sendo desenvolvida; por exemplo, descrições de hardware e banco de dados. Requisitos de hardware definem as configurações mínimas e opcionais do sistema. Os requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e o relacionamento entre os dados.
Índice	Podem ser incluídos vários índices ao documento. Do mesmo jeito que um índice alfabético, pode ter um índice de diagramas, índice de funções, e assim por diante.

Quadro 1: Exemplo de estrutura de um documento de requisitos.
 Fonte: Sommerville (2011, p. 93, tradução nossa)

O nível de detalhamento do documento de requisitos depende do tipo de software a ser desenvolvido e da abordagem a ser adotada. O nível de detalhes do documento de requisitos de um projeto com abordagem evolucionária pode ser menor que em outros projetos (SOMMERVILLE, 2011).

O diagrama de casos de uso é uma ferramenta que permite documentar requisitos de software (OMG, 2015). “Um caso de uso é uma especificação de

comportamento. Uma instância de um caso de uso se refere a uma ocorrência do comportamento emergente em conformidade com o caso de uso correspondente” (OMG, 2015, p. 679, tradução nossa).

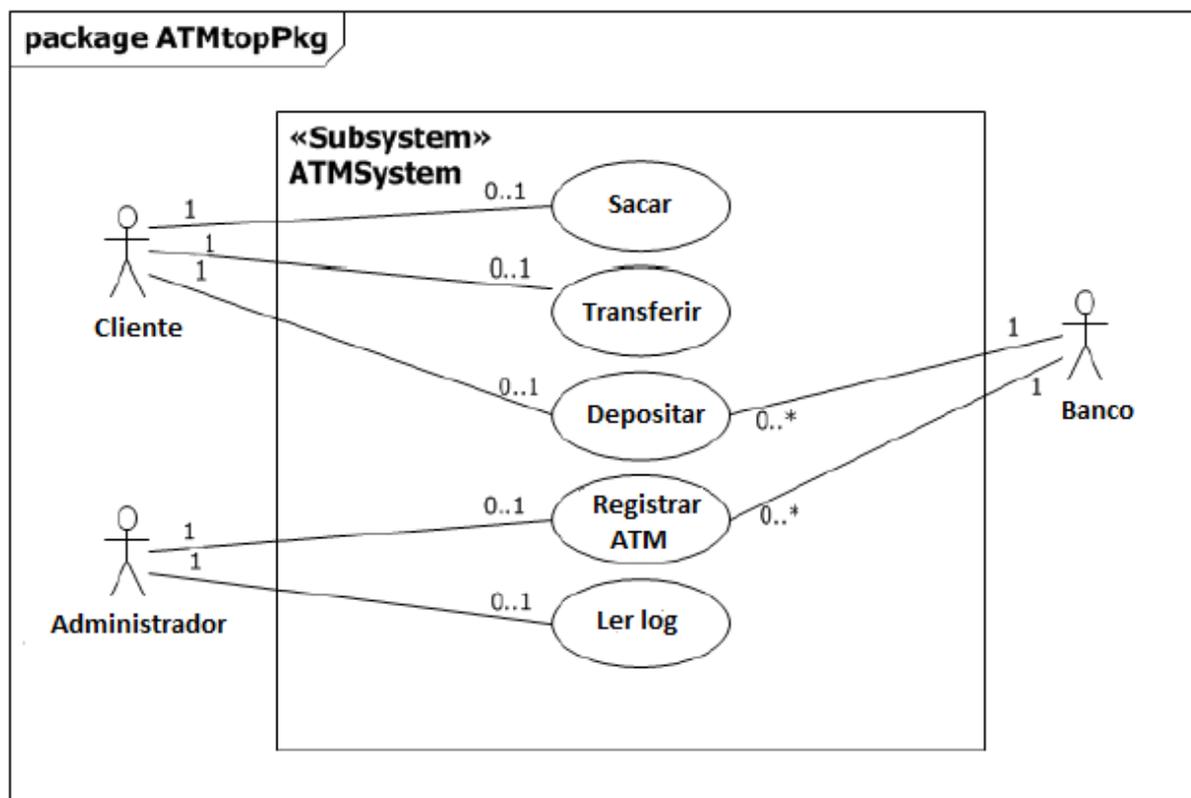


Figura 1: Diagrama de classes de um pacote que possui um conjunto de Casos de Uso, Atores, e um subsistema.

Fonte: OMG (2015, p. 683, tradução nossa)

Assim como o diagrama de casos de uso o diagrama de atividades é um artefato que pode ser usado no momento de especificar requisitos de software, pois, segundo Bell (2003) possibilita modelar o fluxo de um processo definindo todas as ações da atividade. Ainda segundo Bell (2003), o diagrama de atividade pode ser usado para especificar de forma mais detalhada um caso de uso.

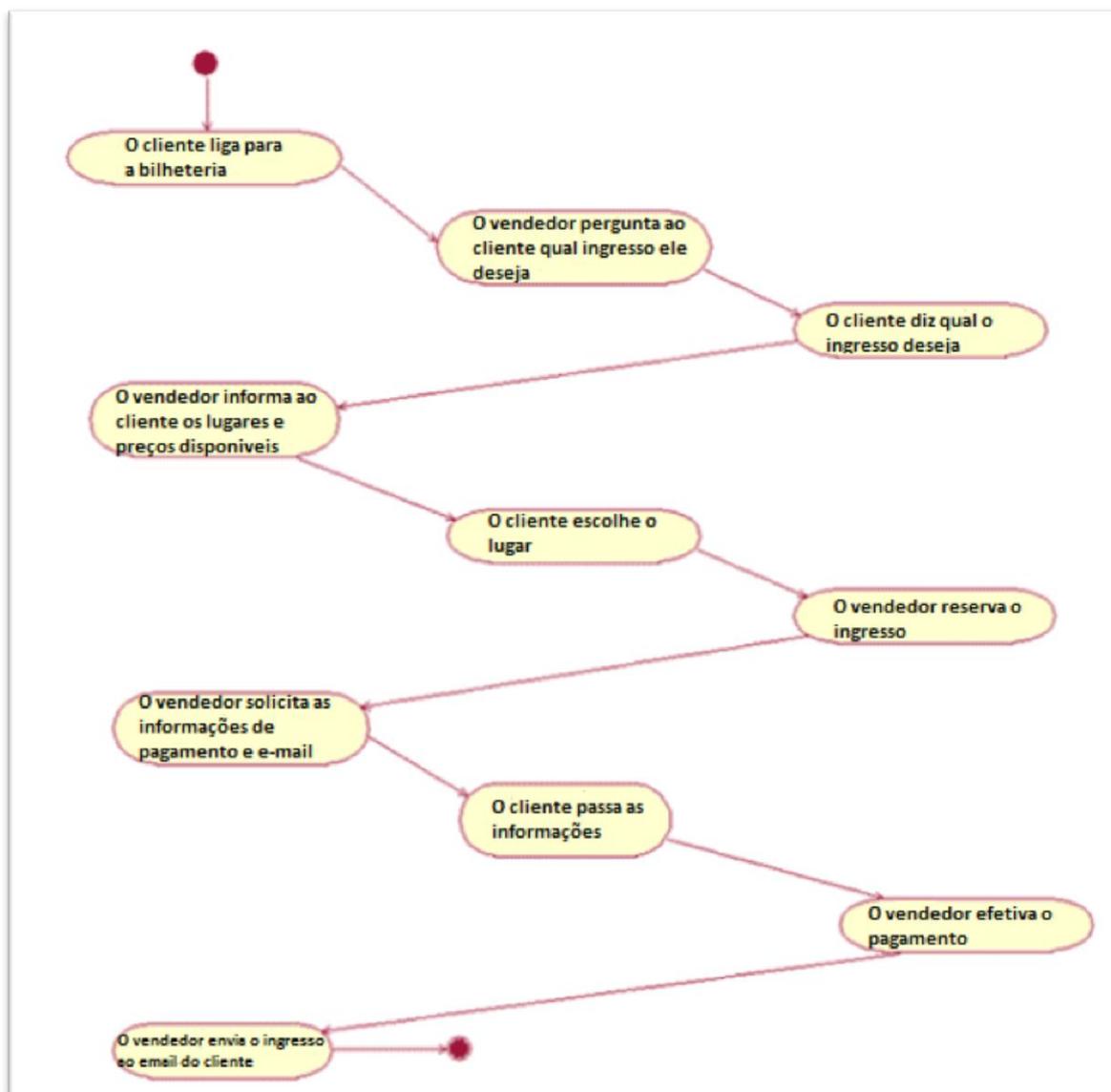


Figura 2: Diagrama de atividade que faz a sequência de ações fácil de entender.
Fonte: Bel (2003, p. 4, tradução nossa)

2.2 Inteligência coletiva

2.2.1 Conceitos e objetivos

Conhecimento trata-se de um fenômeno pessoal, que não pode ser capturado ou compartilhado e surge a partir de processos mentais de compreensão e aprendizado (BLANCK e JANISSEK-MUNIZ, 2014). Sendo o conhecimento um fator determinante de vantagem e posição competitiva para organizações, esse deve ser gerido como um ativo organizacional (BLANCK e JANISSEK-MUNIZ, 2014). A inteligência coletiva busca, a partir de meios tecnológicos, identificar as aptidões individuais das pessoas e coordená-las em uma mobilização de esforços a favor da comunidade (BEMBEM e SANTOS, 2013). Uma das premissas da inteligência coletiva é que “não existe saber que não possa ser transmitido; ninguém é conhecedor do saber absoluto, assim como não existem pessoas desprovidas de saber” (LOZZI, 1996, p. 72).

A tecnologia da informação e comunicação tem enraizada em si os conceitos de inteligência coletiva, o que pode ser facilmente comprovado pela *WEB 2.0*, onde qualquer pessoa pode participar ativamente tanto na produção quanto no consumo de conteúdo (BEMBEM e SANTOS, 2013). A partir do *ciberespaço*, nome dado ao espaço virtual criado pela tecnologia da informação, pode ocorrer a união dos conhecimentos dos indivíduos, mantendo-os interligados independentemente de estarem ou não geograficamente próximos (BEMBEM e SANTOS, 2013).

A Internet evoluiu a ponto de hoje possuir sua própria cultura, a *ciber cultura*, com características e propriedades criadas pela comunidade de internautas (XIBERRAS, 2010). O saber subjetivo, saber individual como o encontrado em *blogs*, tem sido cada vez mais reconhecido em sua legitimidade (XIBERRAS, 2010). A Internet tem rumado em direção de ser um espaço de concentração do conhecimento humano, além de tornar este saber mais acessível a pessoas com acesso à rede mundial de computadores (XIBERRAS, 2010).

2.2.2 Inteligência estratégica antecipativa coletiva

A inteligência estratégica antecipativa coletiva é um conceito de aplicação da inteligência coletiva em ambiente empresarial que busca apoiar o processo de decisão em organizações (BLANCK e JANISSEK-MUNIZ, 2014). A partir de ciclos de coleta, interpretação de dados e difusão dos resultados obtidos a inteligência estratégica antecipativa coletiva cria, de forma colaborativa, conhecimento dentro de uma organização (FERREIRA e JANISSEK-MUNIZ, 2013).

A tarefa inicial é definir o domínio de aplicação da metodologia, selecionar os envolvidos, definir os atores e por fim o tema em que se deseja atuar (FERREIRA e JANISSEK-MUNIZ, 2013), conforme mostra a Figura 3.

	MARKETING	RECURSOS HUMANOS	INOVAÇÕES	MODA (TENDÊNCIA)	MERCADO CALÇADISTA	MEIO AMBIENTE	CONCORRÊNCIA	PRODUTOS	MÍDIAS DIGITAIS	PODER DE COMPRA
DATELLI	X	X	X	X	X	X	X	X	X	X
ALEANZA			X	X	X	X	X	X		
FRANQUIA VILA VELHA	X			X	X		X	X	X	X
FRANQUIA GRAMADO	X			X	X		X	X	X	X
GOVERNO						X				X
CLIENTES	X			X				X	X	X
ADMINISTRAÇÃO					X		X	X		
SISTEMA BANCÁRIO					X					X
COMÉRCIO VAREJISTA	X	X	X	X	X	X	X	X	X	X
INDÚSTRIA CALÇADISTA	X		X	X	X	X	X	X		

Figura 3: Matriz representando os atores e os temas em que irão atuar.
Fonte: Ferreira e Janissek-Muniz (2013, p. 5)

A segunda fase contempla a coleta de informações organizacionais e ocupa a maior parte do trabalho de campo (FERREIRA e JANISSEK-MUNIZ, 2013). As informações devem ser obtidas a partir de captadores voluntários e esses devem ser escolhidos levando-se em conta o nível de interesse no domínio da aplicação e afinidade com os tipos de informações (FERREIRA e JANISSEK-MUNIZ, 2013). As informações captadas são chamadas de sinais fracos, por comumente serem ambíguas, imprecisas, fragmentadas e incompletas (FERREIRA e JANISSEK-MUNIZ, 2013).

Após a coleta de informações inicia-se a etapa de seleção das informações (FERREIRA e JANISSEK-MUNIZ, 2013). Nesse momento são definidos os critérios

de escolha e as informações devem ser filtradas conforme a relevância (FERREIRA e JANISSEK-MUNIZ, 2013). As informações filtradas devem ser armazenadas para formação de uma base de conhecimento, denominada de memória, e para posterior repasse aos interessados (FERREIRA e JANISSEK-MUNIZ, 2013).

As informações selecionadas são usadas na fase chamada de criação coletiva de sentido, que é a última etapa da inteligência estratégica antecipativa coletiva, pois é nessa fase que as informações passam por um processo de interpretação coletiva (FERREIRA e JANISSEK-MUNIZ, 2013). Ferreira e Janissek-Muniz (2013) sugerem a utilização da ferramenta *puzzle* para organização das informações em grupos de informações correlatas. Nessa etapa busca-se criar conhecimento a partir da análise dos sinais fracos e com isso atingir os objetivos definidos na fase de escolha do domínio da aplicação (FERREIRA e JANISSEK-MUNIZ, 2013).

Além das quatro etapas apresentadas, a metodologia inteligência estratégica antecipativa coletiva também possui um papel conhecido como “animador” que é desempenhado por uma pessoa que tem a função de facilitar, coordenar e gerenciar o armazenamento das informações coletadas durante todo o processo de aplicação dessa metodologia (FERREIRA e JANISSEK-MUNIZ, 2013).

2.2.3 A ferramenta *Puzzle*

A ferramenta *puzzle* tem esse nome devido a uma analogia com os jogos de quebra cabeça, pois o *puzzle* permite construir conhecimento a partir de fragmentos de informação (ROUIBAH e OULD-ALI, 2002). Segundo Rouibah e Ould-Ali (2002) existe mais de uma metodologia de criação coletiva de sentido denominada *puzzle*. Rouibah e Ould-Ali (2002) propuseram um *framework puzzle* conforme, descrito a seguir:

O *framework puzzle* é subdividido em sete etapas que abrangem desde a coleta até a proposta de ação. Todo o *framework* é baseado em interpretações de sinais fracos e na construção de conhecimento de forma colaborativa. O *framework puzzle* proposto por Rouibah e Ould-Ali (2002) possui as seguintes características:

- Etapa 1: coleção de dados: Nessa etapa são colecionadas as informações que são os sinais fracos. O ambiente de coleta de dados pode ser limitado definindo-se fontes de informação estratégicas, atores e temas.
- Etapa 2: enriquecimento de dados: Todos os funcionários da empresa podem contribuir com o enriquecimento de sinais fracos. Ao passar por várias pessoas o sinal fraco pode se beneficiar das diferentes interpretações.
- Etapa 3: categorização dos sinais fracos: Os sinais fracos, após serem enriquecidos devem ser categorizados e classificados conforme o tema.
- Etapa 4: criação dos *puzzles*: Algumas informações são selecionadas dos temas para então formar os *puzzles* a partir da combinação de sinais fracos em mapas visuais. Segundo Rouibah e Ould-Ali (2002, p. 142, tradução nossa) o *puzzle* é: “um mapa visual, focado em um ator específico, onde os nós são pequenas sentenças (frases) correspondendo aos sinais fracos; e as pontas são conexões de razão (confirmação, contradição, casualidade) que conectam os diferentes nós”.

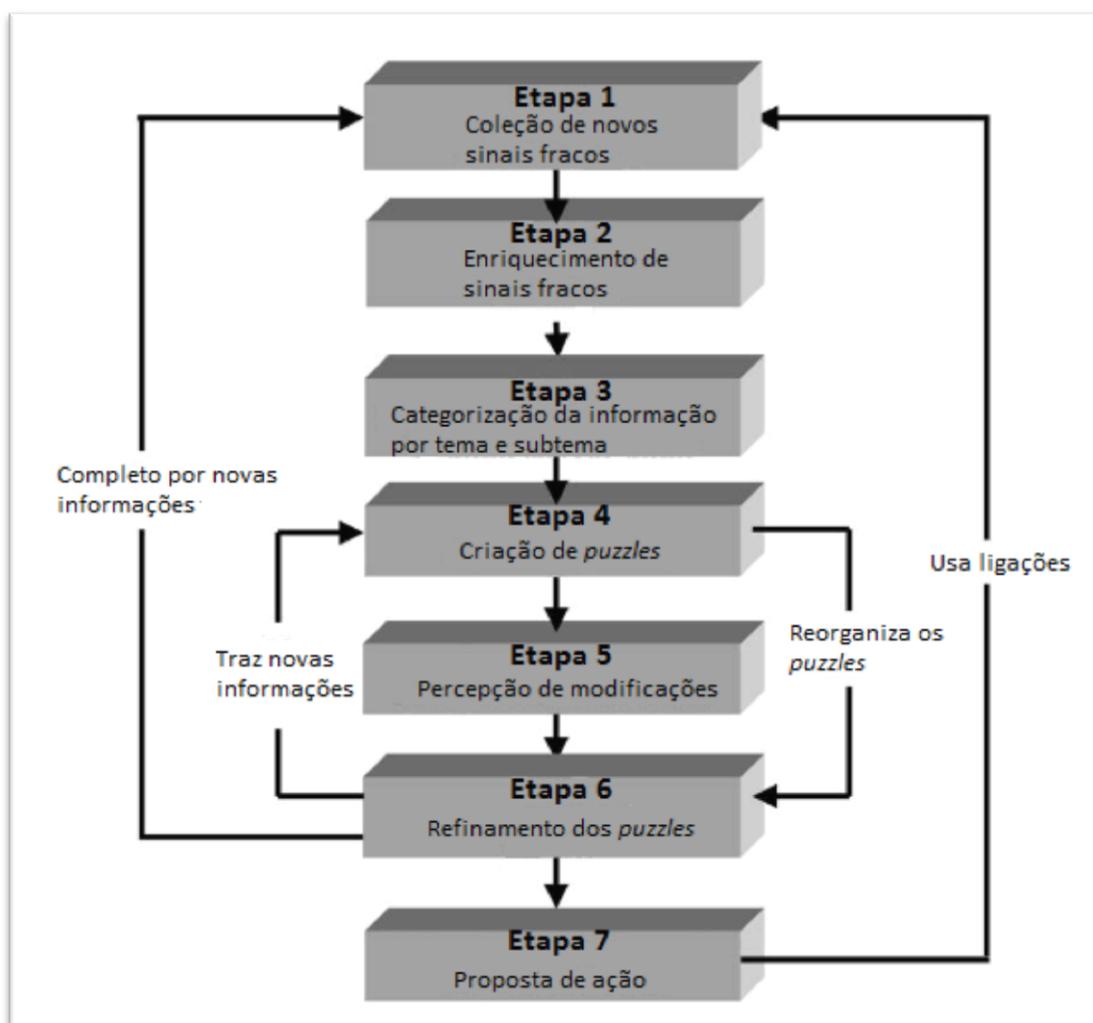


Figura 4: Framework Puzzle.

Fonte: Rouibah e Ould-Ali (2002, p. 141, tradução nossa)

- Etapa 5, percepção de modificações: O primeiro *puzzle* criado é chamado de G_0 . Essa etapa trata de modificações ocorridas nesse *puzzle*. Os seguintes passos devem ser executados:
 - Remover informações obsoletas.
 - Subdividir informações ricas.
 - Agrupar duas informações do G_0 quando elas forem iguais ou uma delas for redundante.
 - Adicionar um vínculo entre duas informações em que uma tenha algum vínculo com a outra.
 - Excluir um vínculo entre duas informações.

- Etapa 6, Refinamento dos *puzzles*: Sempre que novos sinais fracos forem coletados, os *puzzles* devem ser redefinidos. Isso ocorre, pois novas informações podem modificar informações existentes.
- Etapa 7, Proposta de ação: As informações levantadas ajudam o usuário a raciocinar. Esse processo de racionalização é conduzido por questionamentos e argumentos. Ações possíveis são:
 - Verificar a coerência entre os elementos.
 - Orientar a busca de informações faltantes
 - Adicionar novas informações a itens adjacentes que necessitam de validação imediata.

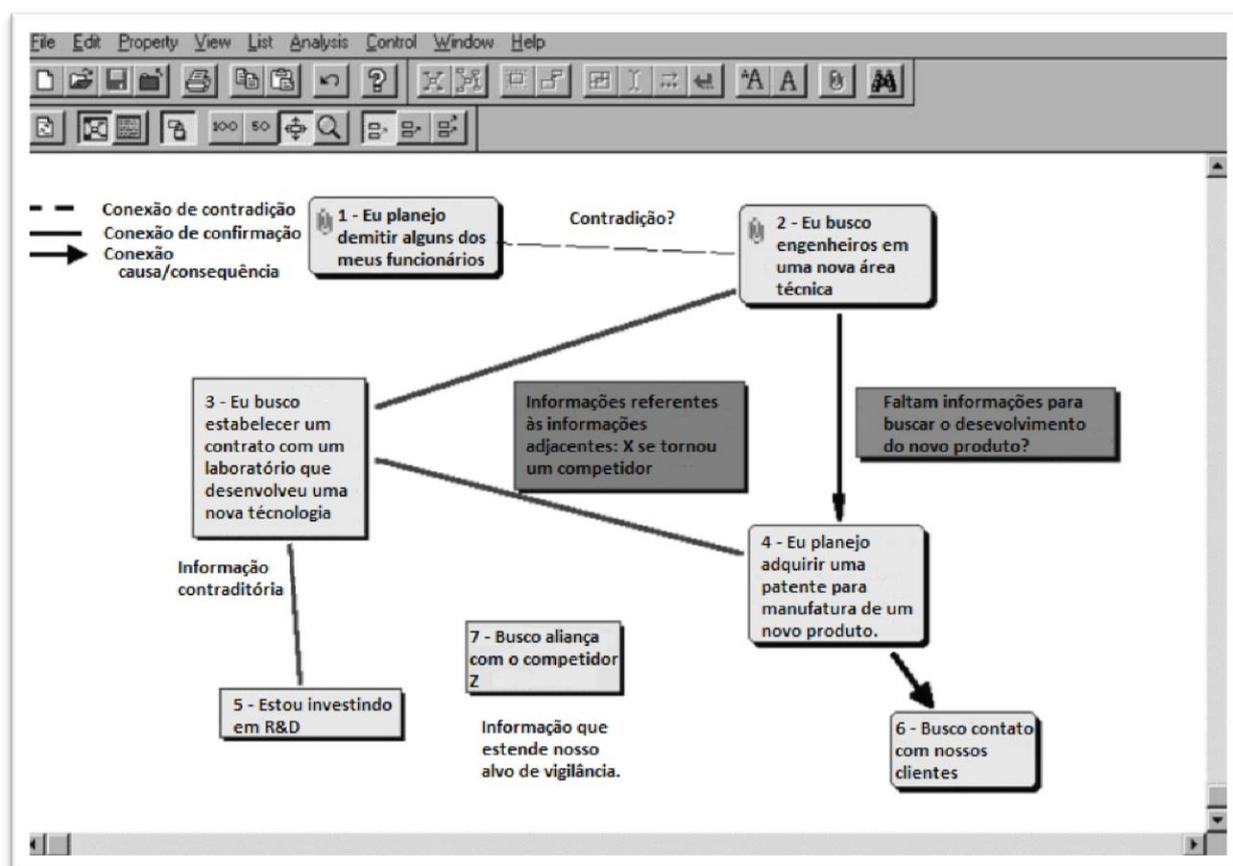


Figura 5: Exemplo de *Puzzle*

Fonte: Rouibah e Ould-Ali (2002, p. 144, tradução nossa)

O resultado da aplicação do *puzzle* é um diagrama contendo as informações adquiridas. A ferramenta é baseada na construção desse diagrama, que apresenta

de forma simplificada e organizada o conhecimento obtido a partir do estudo e refinamento dos sinais fracos. Podemos identificar em um *puzzle* alguns quadros contendo informações além de ligações entre eles, conforme pode ser visto na Figura 5.

O diagrama gerado como resultado da utilização da ferramenta *puzzle* pode ser usado, por exemplo, como fonte de conhecimento para tomada de decisões estratégicas, estudo de mercado ou desenvolvimento de um novo produto. O tipo das informações contidas em um *puzzle* é definido pelo seu tema e, conseqüentemente, sua utilidade é definida pelo tema escolhido.

3 APRESENTAÇÃO E ANÁLISE DE RESULTADOS

3.1 Visão geral do modelo proposto

O processo de levantamento de requisitos proposto por esse trabalho apresenta um modelo de utilização de inteligência coletiva aplicada na engenharia de requisitos, que foi desenvolvido a partir do estudo do processo de engenharia de requisitos e inteligência coletiva.

O modelo pode ser subdividido em: papéis, ciclos e artefatos. Os papéis do modelo representam grupos de participantes do processo de levantamento de requisitos, sendo que cada papel possui suas próprias atribuições e responsabilidades. Os ciclos representam a disposição ordenada das etapas do processo, sendo que cada etapa possui um objetivo e tarefas a serem executadas. O modelo prevê dois ciclos, sendo um para as etapas de levantamento de requisitos e um ciclo contínuo com atividades comuns a todo o processo e que não estão diretamente ligadas ao levantamento de requisitos. Também são definidos, para o processo, alguns artefatos que de maneira geral são documentos gerados a partir da execução de algumas das tarefas do modelo.

Cada item desse processo foi definido com um objetivo único e inspirado em papéis, tarefas ou artefatos já existentes em engenharia de requisitos e inteligência coletiva. Portanto esse pode ser considerado uma adaptação e união das tarefas de engenharia de software, da inteligência estratégica antecipativa coletiva e da ferramenta *puzzle* com objetivo de criar uma nova forma de levantamento de requisitos em projetos de desenvolvimento de software.

3.2 Papéis

Para o modelo proposto foram definidos quatro papéis a serem desempenhados durante o decorrer do processo de levantamento de requisitos com

inteligência coletiva. Esses papéis foram propostos visando a definir responsáveis pelas atividades do processo tendo cada papel um objetivo único. São eles:

3.2.1 Colaborador

O papel de colaborador foi criado baseando-se nos captadores voluntários da inteligência estratégica antecipativa coletiva (BLANCK e JANISSEK-MUNIZ, 2014) e no ambiente de coleta de dados da ferramenta *puzzle* (ROUIBAH e OULD-ALI 2002). As pessoas que ocupam esse papel são as fontes de informações para o processo de levantamento de requisitos proposto por esse trabalho.

É dado o papel de colaborador a todos que possam contribuir com informações para o levantamento dos requisitos do sistema. Eles formam a principal fonte de informações do processo e possuem a responsabilidade de registrar os conhecimentos relevantes ao projeto, registrar possibilidades de aperfeiçoamento dos processos e propor melhorias no sistema. São, em geral, pessoas com conhecimento de negócio oriundo de experiência adquirida a partir da vivência operacional de suas funções. Qualquer pessoa pode ser um colaborador, desde que possa contribuir com seu conhecimento para o levantamento de requisitos.

A responsabilidade das pessoas que desempenham esse papel está relacionada à manipulação das informações que darão origem aos requisitos de software. Elas devem trabalhar em conjunto para formar uma base de conhecimento sobre o negócio e o software a ser desenvolvido.

Apesar das pessoas com esse papel possuírem a responsabilidade de manter as informações que serão usadas para o levantamento dos requisitos, elas não possuem autonomia para aceitar ou rejeitar um requisito de software, devido à possibilidade de divergência de opiniões, entre os colaboradores, relacionadas aos requisitos de software. Apenas negociadores possuem autonomia para aceitar ou rejeitar um requisito de software.

3.2.2 Animador

O papel de “animador”, definido para o modelo de processo para levantamento de requisitos proposto por esse trabalho, foi adaptado tendo como base o “animador” do conceito de inteligência estratégica antecipativa coletiva. Além da atribuição de incentivador para a participação no processo definido pela ferramenta *puzzle*, o “animador” passa a ser responsável pela gestão de *stakeholders*.

Esse papel tem como principal objetivo atuar no incentivo à participação das pessoas no processo de levantamento de requisitos. Os “animadores” possuem a missão de identificar potenciais colaboradores, garantir que esses tenham acesso às ferramentas usadas no processo de levantamento de requisitos e buscar a participação de todos que possam contribuir com informações relevantes para construção do sistema de informação.

A primeira responsabilidade dos animadores é identificar todos os *stakeholders* do projeto e qualificá-los quanto à natureza do interesse e possibilidade de contribuição. Um exemplo de qualificação de *stakeholders* poderia ser quanto ao envolvimento com o negócio (por exemplo: a pessoa atua ativamente no negócio ou apenas é informada sobre sua situação?), quanto às atividades executadas (por exemplo: a pessoa possui interesse em funções do setor de compras, vendas ou qualquer outro que seja contemplado pelo projeto?) ou qualquer outra forma de qualificação considerada relevante pelo “animador”. A atividade de identificação dos participantes do processo é prevista na primeira etapa da inteligência estratégica antecipativa coletiva e atribuída aos “animadores” para o modelo de levantamento de requisitos definido por este trabalho.

Uma vez identificados os *stakeholders* que terão papel de colaboradores no processo de levantamento de requisitos, o “animador” deverá certificar-se que todos eles possuem acesso às ferramentas necessárias para que possam, de fato, contribuir com seus conhecimentos para criação de uma base de conhecimento. Esta é uma atribuição previamente definida para os “animadores” da inteligência estratégica antecipativa coletiva.

A principal função desse papel está em incentivar os colaboradores a contribuírem com seu conhecimento, atividades e processos que devem ser executados durante as atividades organizacionais. Os “animadores” devem ajudar os colaboradores a entenderem como eles podem auxiliar na criação da base de conhecimento sobre o negócio, ajudando-os na descoberta de habilidades individuais relevantes ao negócio, sanando dúvidas relacionadas ao processo de levantamento de requisitos e garantindo que todos entendem o valor criado por sua contribuição ao projeto.

Como atribuições específicas desse papel, os animadores atuam em todas as fases do levantamento de requisitos como pessoas responsáveis por manter o processo em constante funcionamento.

3.2.3 Negociador

O papel de negociador não foi baseado em nenhum papel da engenharia de requisitos, inteligência estratégica antecipativa coletiva ou da ferramenta *puzzle*. Ao analisar as tarefas da engenharia de requisitos de Pressman (2011), percebeu-se a necessidade de definir pessoas responsáveis por centralizar a responsabilidade de atuar na tarefa de negociação proposta por Pressman (2011). A necessidade de definir alguém responsável pelas negociações ocorre, também, por não haver um papel responsável por auxiliar os analistas de requisitos a sanar suas dúvidas caso eles enfrentem dificuldades na hora de montar os *puzzles*. A falta do papel de negociador poderia dificultar a tomada de decisões relacionadas aos requisitos, uma vez que não haveria responsáveis representantes dos temas do projeto de software.

Negociador é a pessoa responsável por negociações de escopo e gestão de conflitos no projeto. Recomenda-se haver um negociador por tema do software e esse terá autonomia para realizar decisões relacionadas ao tema de sua responsabilidade. Sendo o negociador um papel de decisão e gestão de conflitos, espera-se que haja apenas um negociador por tema como forma de evitar conflitos entre negociadores.

É importante que o negociador seja uma pessoa com grande conhecimento sobre o negócio, uma vez que ele será responsável por tomar decisões relacionadas aos requisitos pertencentes ao tema pelo qual é responsável. Ele também será a pessoa com a qual os engenheiros de requisitos podem tirar dúvidas pontuais, negociar o escopo, entregar as propostas de requisitos para aprovação, comunicar-se em caso de requisitos conflitantes ou qualquer outro assunto relacionado à tomada de decisões sobre os requisitos de software.

Negociadores também devem buscar, em conjunto, formas de integrar os requisitos quando impactarem mais de um tema, procurando identificar a melhor solução possível para o problema. Em situações em que o requisito tenha impacto em mais de um tema os negociadores responsáveis por cada tema devem buscar uma solução que atenda a todos, sem que haja prejuízo para nenhum dos lados.

O negociador também deve auxiliar os analistas de requisitos no processo de criação de sentido coletivo, utilizando a ferramenta *puzzles*, e validar os requisitos certificando-se de que eles atendem ao negócio. Ele pode ajudar os analistas a entenderem melhor os sinais fracos e como eles se relacionam para criação de sentido do ponto de vista do tema.

3.2.4 Analistas de requisitos

O papel de analista de requisitos foi incorporado a partir da engenharia de requisitos. O analista de requisitos é a pessoa responsável por realizar o levantamento de requisitos de um software. No modelo de processo de engenharia de requisitos com inteligência coletiva proposto por esse trabalho, o engenheiro de requisitos possui a mesma responsabilidade que em processos de levantamentos de requisitos tradicionais. São pessoas com conhecimentos relacionados ao levantamento de requisitos e desenvolvimento de software.

A missão dos analistas de requisitos é organizar os sinais fracos, coordenar o trabalho coletivo para o levantamento de requisitos de software e analisar os *puzzles* em busca de requisitos de software.

O analista de requisitos tem o papel guia para o processo de levantamento de requisitos. Ele é o principal responsável pela criação dos *puzzles* e pela extração dos requisitos a partir das informações coletadas. Essas atribuições são dadas ao analista de requisitos, pois ele é a pessoa que detém o conhecimento sobre o levantamento de requisitos.

A partir da sinalização de inconsistências, deficiências e incompatibilidades dos sinais fracos o analista de requisitos pode coordenar a coleta de informações. Os analistas de requisitos também podem coordenar a coleta de informações a partir da solicitação direta de informações a negociadores e animadores que devem providenciar essas informações junto aos colaboradores.

Apesar de a colaboração do negociador durante a criação dos *puzzles* ser muito importante, essa tarefa é de total responsabilidade do analista de requisitos. Ele deve montar os *puzzles* e disponibilizá-los para que os colaboradores e negociadores possam auxiliar no seu refinamento.

3.3 O ciclo do processo

Foi definido um ciclo para o processo de levantamento de requisitos com inteligência coletiva. Esse ciclo ajuda a guiar os envolvidos durante o processo, além de ajudar a separar as tarefas a serem executadas de forma ordenada e facilitando a compreensão do processo.

As etapas do ciclo de levantamento de requisitos de software com inteligência coletiva foram propostas conforme as tarefas da engenharia de requisitos definidas por Pressman (2011). Para cada tarefa da engenharia de requisitos levantada por Pressman (2011) foi planejada uma etapa no modelo de levantamento de requisitos proposto por esse trabalho. As etapas do processo de levantamento de requisitos com inteligência coletiva mantêm as mesmas características das tarefas originalmente definidas por Pressman (2011).

Para que cada etapa do processo de levantamento de requisitos com inteligência coletiva possa ter seu objetivo cumprido, foram definidas tarefas específicas para cada uma. As tarefas foram definidas conforme o conceito de

inteligência estratégica antecipativa coletiva (BLANCK e JANISSEK-MUNIZ, 2014) e a ferramenta *puzzle*(ROUIBAH e OULD-ALI 2002).A Figura 6 representa graficamente o ciclo de etapas do processo proposto por este trabalho.

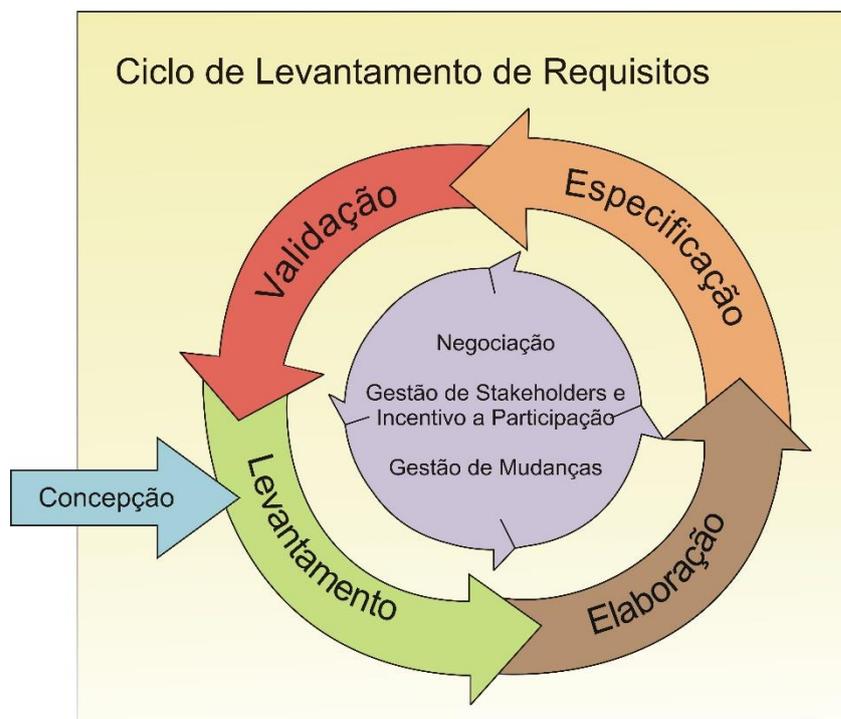


Figura 6: Representação do ciclo de levantamento de requisitos com inteligência coletiva.
Elaboração própria

3.3.1 Concepção

Da mesma forma que a tarefa de concepção definida por Pressman (2011), essa tarefa trata dos motivos que dão origem ao projeto e do objetivo definido para ele. Ao se iniciar um projeto de software deve-se definir o objetivo, isso é, o problema a ser resolvido, a abrangência do sistema na organização e o resultado esperado. Esse objetivo deve ser perseguido durante todo o levantamento de requisitos de software e conseqüentemente o desenvolvimento do sistema. A necessidade de definir um objetivo para o projeto foi identificada ao analisar a tarefa de concepção

de Pressman (2011) que deve tratar dos motivos que dão início a um projeto software.

A definição do objetivo não é de responsabilidade de nenhum dos papéis definidos para o processo, mas uma decisão organizacional que ocorre antes mesmo do início do projeto de software. Essa tarefa não foi atribuída a nenhum papel do modelo proposto, pois ela deve ocorrer antes do início do projeto.

Um exemplo de objetivo de projeto seria: construir um sistema ERP, que integre os setores de compras e vendas para eliminar atividades redundantes, garantir a consistência da informação e dos fluxos produtivos e reduzir custos operacionais. Nesse exemplo podemos identificar alguns problemas que devem ser resolvidos ou mitigados, os setores atendidos e o cenário desejado com a implantação do software. Essas informações são importantes para o processo, pois guiam os participantes em suas atividades.

Diferentemente do conceito original para tarefa de concepção definida por Pressman (2011), que não trata da formalização do motivo que deu origem ao projeto de software, para o modelo de engenharia de requisitos definido por este trabalho recomenda-se essa formalização haja visto a importância de explicitar o que se espera do projeto, uma vez que atender a esse objetivo pode ser considerado um requisito.

As tarefas para essa etapa do processo são:

- **Identificar uma necessidade:** nessa tarefa uma necessidade, problema ou oportunidade de negócio é identificada e, a partir dela a ideia de construção de um software. Essa tarefa é derivada da especificação original para a tarefa de concepção da engenharia de requisitos definida por Pressman (2011), que busca tratar os motivos que dão origem a um projeto de software.
- **Definir o objetivo do projeto:** recomenda-se que seja definido o objetivo do projeto. O objetivo ajuda a guiar o levantamento de requisitos e conseqüentemente o desenvolvimento do novo software. O objetivo do projeto pode ser encarado como uma definição formal do que se espera da sua implantação. Esta tarefa foi acrescentada ao processo para

formalizar os motivos que dão origem a um projeto de software durante a tarefa de concepção do projeto. O objetivo do projeto pode ser acrescido, também, à base de conhecimento do projeto junto aos demais sinais fracos, uma vez que pode dar origem a requisitos de software.

- **Iniciar o projeto:** após identificada a necessidade de construção do novo software é iniciado um novo projeto de desenvolvimento. Esse projeto, se aceito pela organização, pode iniciar a fase de levantamento de requisitos.

3.3.2 Levantamento

Baseada na tarefa de levantamento de requisitos proposta por Pressman (2011) essa é a etapa definida para levantamento dos sinais fracos. Assim como definido originalmente por Pressman (2011) o foco dessa etapa é a captação de informações.

Os colaboradores, que atuam como os captadores voluntários da inteligência estratégica antecipativa coletiva, são os principais responsáveis pela execução das atividades de levantamento. Eles devem, voluntariamente, atuar em conjunto para construção de uma base de conhecimento tendo sempre em mente o objetivo definido para o projeto. Os sinais fracos devem ser incluídos em um sistema de captação de informações para que possam ficar à disposição de todos os colaboradores e para que esses possam enriquecer essas informações durante a criação dos *puzzles*. Conforme definição da inteligência estratégica antecipativa coletiva, a captação de informação deve ocorrer de forma voluntária. Os colaboradores não devem se sentir coagidos ou forçados a contribuir com a construção da base de conhecimento, mas devem ser motivados a realizar essa contribuição.

Segundo os conceitos da inteligência estratégica antecipativa coletiva e ferramenta *puzzle*, todo tipo de informação pode ser útil para o levantamento de requisitos e nenhuma deve ser desprezada no momento da captação dos sinais

fracos, pois as informações devem ser filtradas durante a construção dos *puzzles*. Os colaboradores devem ser incentivados a alimentar a base de conhecimento com qualquer informação, mesmo que em um primeiro momento essa pareça ser óbvia ou irrelevante.

Além da coleta de informações nessa etapa também são feitas as definições dos temas. Os temas são divisões lógicas e de alto nível que façam algum sentido para o software e devem seguir a especificação original da ferramenta *puzzle* proposta por Rouibah e Ould-Ali (2002). Eles podem ser módulos do sistema (como módulo de compras e vendas), subsistemas (como loja virtual e sistema administrativo), ou qualquer outra divisão considerada relevante. Os temas facilitam o relacionamento dos sinais fracos no momento da criação dos *puzzles*.

Para cada tema deve ser escolhido um e apenas um negociador. A escolha do negociador ocorre conforme preferência da empresa. Podendo ser desde uma votação entre os colaboradores até uma indicação da gerência. A única restrição para a escolha do negociador é que seja uma pessoa com sólido conhecimento sobre o tema que vai representar. A definição de um negociador por tema surgiu como uma forma de preencher uma lacuna identificada ao adaptar a engenharia de requisitos a uma abordagem voltada à inteligência coletiva. Ao verificar que o processo de engenharia de requisitos seria prejudicado, por não haver uma pessoa responsável por centralizar decisões sobre os temas e os requisitos, optou-se por criar o papel de negociador e torná-lo responsável pelos temas.

A etapa de levantamento possui algumas tarefas que devem ser cumpridas para que o processo funcione corretamente. As tarefas previstas para essa etapa são:

- **Definir temas:** essa tarefa foi incorporada da inteligência estratégica antecipativa coletiva, e é nela que devem ser definidos os temas do software. Como já mencionado, esses temas são subdivisões lógicas do software a ser construído. Pode ser usado qualquer critério desejado para essa divisão. Os temas são criados pelos analistas de requisitos em conjunto com a alta gestão do cliente e colaboradores do projeto. Exemplos de critérios para definição dos temas poderiam ser: módulos do sistema, departamentos do cliente que usarão o software e subsistemas. Cada tema dará origem a um *puzzle* construído a partir de

seus sinais fracos. É possível que sejam descobertos relacionamentos entre dois ou mais temas durante a análise dos sinais fracos e isso deve ficar explícito na construção dos *puzzles*.

- **Selecionar negociadores:** Para cada tema definido deve ser indicado um responsável que assumirá o papel de negociador. O negociador será responsável pelas negociações necessárias durante o processo de levantamento de requisitos, tendo cada negociador autonomia para a tomada de decisões relacionadas ao tema que representa. O cliente tem liberdade para nomear os negociadores conforme desejado. Porém como já salientado, o negociador deve conhecer bem o tema que representa para que possa tirar dúvidas dos analistas de requisitos além de validar os requisitos levantados. Essa tarefa não tem origem nos conceitos de inteligência coletiva ou engenharia de requisitos estudados e foi acrescentada ao modelo proposto por este trabalho para incluir a seleção dos negociadores.
- **Construção da base de conhecimento:** Essa tarefa, análoga a tarefa de coleta de informações da inteligência estratégica antecipativa coletiva, consiste em incluir os sinais fracos na base de conhecimento do projeto. Os colaboradores incluem os sinais fracos na base de conhecimento para que essas informações fiquem disponíveis a todos os *stakeholders* do projeto. Essa pode ser considerada a principal atividade do processo de levantamento de requisitos com inteligência coletiva, pois é a partir das informações compartilhadas pelos colaboradores que os requisitos do software serão levantados.
- **Enriquecer os sinais fracos:** Os sinais fracos adicionados à base de conhecimento devem ficar à disposição de todos os colaboradores para que esses possam enriquecer esses sinais fracos com informações adicionais. Todos devem poder enriquecer qualquer sinal fraco, independentemente do tema a que estiver vinculado. O enriquecimento dos sinais fracos deve contribuir para uma maior assertividade e

detalhamento dos requisitos de software. Essa tarefa já é prevista pela ferramenta *puzzle* e foi incorporada ao modelo proposto por este trabalho.

- **Categorizar os sinais fracos:** Conforme a ferramenta *puzzle*, os sinais fracos devem ser categorizados conforme sua afinidade com os temas definidos para o processo. É possível que o mesmo sinal fraco tenha relação com dois ou mais temas. Os sinais fracos também podem ser subdivididos em categorias e subcategorias durante o levantamento. Porém essa metodologia prevê que essa subdivisão seja feita de forma definitiva na etapa de elaboração dos requisitos.

As tarefas definidas para a etapa de levantamento foram projetadas para suportar o levantamento de informações e dar início à criação dos *puzzles*. Essa é a primeira etapa, do processo sugerido por este trabalho, que realiza a união entre engenharia de requisitos e inteligência coletiva. O levantamento de informações busca atender aos objetivos definidos por Pressman (2011) para a tarefa de levantamento. A criação dos temas e enriquecimento e categorização dos sinais fracos buscam atender e dar início à aplicação da ferramenta *puzzle* com objetivo de levantar informações necessárias para o levantamento de requisitos.

As atividades previstas para a etapa de levantamento de informações podem ocorrer de forma concorrente, isso é, duas ou mais tarefas podem ser executadas ao mesmo tempo. Apesar de o levantamento de informações ocorrer durante todo o período do projeto ele foi definido como uma etapa do processo para que possam ser criados ciclos de levantamento de requisitos. Dessa forma periodicamente os analistas de requisitos podem usar as informações coletadas para iniciar uma rodada de construção ou atualização dos *puzzles* e levantamento dos requisitos para o software. Caso o levantamento de informações fosse tratado como uma etapa contínua seria mais difícil isolar uma versão da base de informações para o levantamento de requisitos, uma vez que ela estaria em constante mudança.

	Alta gestão	Negociador	Colaborador	Analista de requisitos
Definir temas	R		R	R
Selecionar negociadores	R	I	R	
Construção da base de conhecimento		C/A	R	I
Enriquecer os sinais fracos		C/A	R	R/I
Categorizar os sinais fracos		C/A	R	I

Quadro 2: Matriz RACI da etapa de levantamento
Elaboração própria

3.3.3 Elaboração

A etapa de elaboração segue os mesmos princípios da tarefa de elaboração definida por Pressman (2011), porém adaptada à inteligência coletiva. Essa adaptação ocorre dando a essa etapa um foco maior na elaboração dos *puzzles* para, só então, usá-los para elaborar os requisitos de software. Pode-se dizer, então, que essa etapa é fundamentada tanto na tarefa definida por Pressman (2011), quanto na inteligência estratégica antecipativa coletiva (BLANCK e JANISSEK-MUNIZ, 2014) e ferramenta *puzzle* (ROUIBAH e OULD-ALI 2002).

O principal objetivo da etapa de elaboração é a construção dos *puzzles*. Os analistas de requisitos devem criar um *puzzle* para cada tema do projeto. As informações contidas em cada tema são agrupadas conforme seus relacionamentos em categorias e subcategorias.

Categorias e subcategorias em um *puzzle* são grupos de informações correlatas que juntas formam sentido do ponto de vista do tema e do domínio do software a ser desenvolvido. O analista de requisitos é o principal responsável pela elaboração dos *puzzles*, porém negociadores devem colaborar com os analistas de requisitos sempre que esses requisitarem.

Os *puzzles* devem ser refinados tantas vezes quantas forem necessárias até que os analistas de requisitos os percebam como uma representação fiel, não ambígua e consistente das necessidades do cliente. Em geral serão necessárias várias rodadas do processo para que o analista de requisitos possua informações

suficientes para concluir um *puzzle*. Isso ocorre devido à característica incremental, no levantamento de informações, do modelo proposto por este trabalho.

É proposto que em cada rodada de levantamento de requisitos os analistas isolam as informações da base de conhecimento coletadas até o início da etapa de elaboração. Essas informações isoladas são usadas em toda a rodada do processo de levantamento de requisitos e representam os sinais fracos coletados para a construção de um *puzzle*.

Após a separação dos temas em *puzzles*, os analistas de requisitos combinam os sinais fracos para criação de sentido, categorizando as informações conforme seus relacionamentos. Categorias e subcategorias de informações podem ser criadas para ajudar a organizar as informações contidas em cada *puzzle*. Uma forma de categorizar as informações, por exemplo, pode ser criando uma categoria para cada tela do sistema e subtemas para as suas funcionalidades.

Pode-se dizer que o objetivo dessa etapa do modelo proposto é a aplicação da ferramenta *puzzle* usando os sinais fracos coletados durante a etapa de levantamento. Após o término dessa etapa, espera-se que o analista de requisitos tenha criado um ou mais *puzzles*, que servirão como fonte de informações para a identificação e especificação dos requisitos de software.



Figura 7: Mapa mental exemplificando a segregação da base de conhecimento de um sistema ERP em temas, categorias e subcategorias.
Elaboração própria

As atividades a serem realizadas nessa etapa do processo são cinco:

- **Isolar uma versão da base de conhecimento:** A base de conhecimentos de um projeto de software tende a estar em constante mudança. Essa volatilidade da base de conhecimentos pode dificultar o processo de criação dos *puzzles*. Para resolver os problemas relacionados à volatilidade recomenda-se que os analistas de requisitos isolem uma versão da base de conhecimentos para então dar continuidade no processo. Deve-se ressaltar que, sempre que uma versão da base de conhecimentos for isolada para criação dos *puzzles*, isso deve ser feito de forma acumulativa, incluindo também informações já utilizadas em rodadas anteriores do processo, porém não há necessidade de analisar novamente todos os itens. Apenas itens novos ou impactados por novas informações precisam ser analisados. Essa tarefa não foi baseada em nenhuma tarefa da engenharia de requisitos ou inteligência coletiva, mas uma forma identificada para minimizar impactos negativos que podem ser ocasionados pela volatilidade da base de conhecimentos.
- **Criar os *puzzles*:** Para cada tema do sistema deve ser criado um *puzzle*. Os *puzzles* devem seguir a definição original da ferramenta *puzzle* definida por Rouibah e Ould-Ali (2002). Caso um tema já possua um *puzzle* criado em alguma rodada anterior do processo ele poder ser apenas atualizado. Não há necessidade de redefinir os *puzzles* integralmente a cada rodada. Essa tarefa foi incorporada ao modelo proposto seguindo a quarta etapa da ferramenta *puzzle*.
- **Percepção de modificações:** Sinais fracos, quando combinados, podem se tornar informações relevantes para o processo. A partir dessa combinação eles podem, por exemplo, se complementar, reafirmar ou desqualificar. As informações obtidas da combinação dos sinais fracos serão usadas para dar aos analistas de requisitos uma melhor compreensão sobre o negócio e os serviços que o software deve oferecer. Essa tarefa é derivada das etapas de percepção de modificações da ferramenta *puzzle*, que já prega a necessidade de combinar os sinais fracos, para o enriquecimento e descoberta de novas informações.

- **Definir as categorias e subcategorias:** Incorporada, no processo proposto por este trabalho, trata-se da terceira etapa da ferramenta *puzzle*. Essa tarefa é onde as informações obtidas, a partir da combinação dos sinais fracos, devem ser categorizadas conforme seus relacionamentos com o software e com o negócio a ser atendido. Essas informações podem ser classificadas conforme preferência dos analistas de requisitos, buscando sempre atingir o objetivo do projeto. Uma forma possível de classificação das informações em um *puzzle* pode ser criando categorias para as telas e subcategorias para suas funcionalidades. Por exemplo: em um projeto de desenvolvimento de um sistema ERP pode-se criar uma categoria para a tela de pedido de compra com subcategorias para as funcionalidades de inclusão de produtos no pedido ou para emitir a nota fiscal.
- **Refinar os *puzzles*:** Essa também se trata de uma tarefa da ferramenta *puzzle*, incorporada ao processo proposto por este trabalho. Nessa tarefa, os *puzzles* devem ser refinados tantas vezes quantas forem necessárias. É possível que novas informações possam ser identificadas após a montagem dos *puzzles* ou após a inclusão de novos sinais fracos. Sempre que uma nova rodada do processo de levantamento incluir novas informações a um *puzzle* ele deverá ser refinado. O refinamento deve ocorrer sempre que houver qualquer alteração no *puzzle*.

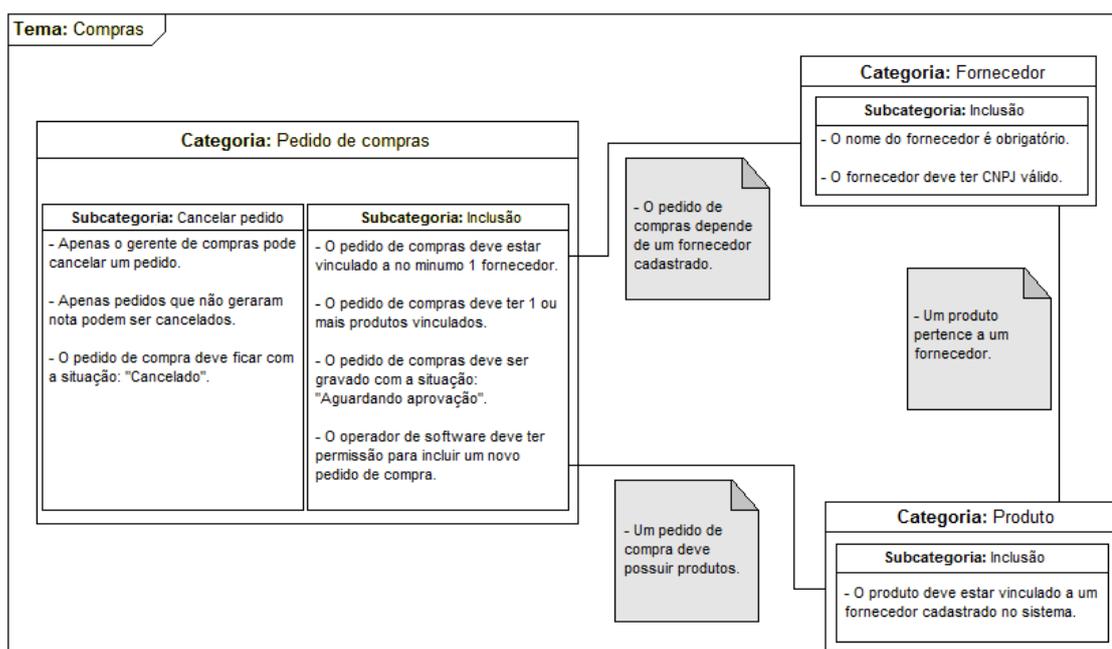


Figura 8: Exemplo de *puzzle* para o tema compras de um ERP.
Elaboração própria

	Colaborador	Negociador	Analista de requisitos
Isolar uma versão da base de conhecimento			R
Criar os <i>puzzles</i>		C/A	R
Combinar os sinais fracos		C/A	R
Definir as categorias e subcategorias		R	R
Refinar os <i>puzzles</i>		R	R

Quadro 3: Matriz RACI da etapa de elaboração
Elaboração própria

3.3.4 Especificação

Esta etapa do processo definido por esse trabalho foi baseada exclusivamente na tarefa de especificação definida por Pressman (2011), porém adaptada para utilizar os artefatos gerados a partir da aplicação da ferramenta *puzzle* como fonte de informações sobre os requisitos de software. É nessa etapa que, segundo Pressman (2011), os requisitos de software devem ser documentados.

Após a etapa de elaboração os *puzzles* estarão prontos para que os analistas de requisitos os possam estudar, para definir os requisitos de software. Essa tarefa é focada nas atividades de identificação e documentação dos requisitos de software.

A partir da análise minuciosa das informações contidas em um *puzzle* o analista de requisitos pode identificar as necessidades a serem atendidas pelo software. O *puzzle* pode ser encarado como uma fonte valiosa de informações sobre o funcionamento da organização, com a vantagem de reunir tudo em um único lugar. Quanto mais maduros forem os *puzzles* mais assertivo será o trabalho de identificação de requisitos e mais próximos da realidade dos clientes eles serão, uma vez que representam as necessidades dos usuários de software, definidas com base em informações que eles mesmos forneceram.

Após identificar um requisito esse deverá ser documentado pelo analista de requisitos de uma forma que ele possa ser enviado para implementação pela equipe de desenvolvimento de software. Essa documentação pode ser feita conforme os padrões definidos pela equipe de desenvolvimento em conjunto com os analistas de requisitos. O documento de especificação de requisitos deve apresentar, de forma fiel, o que será implementado no software e principalmente deve representar um acordo feito com o cliente que torne explícito o que será entregue. São duas as tarefas definidas para essa etapa do processo. São elas:

- **Identificar os requisitos de software:** essa tarefa consiste em analisar os *puzzles* identificados em busca de requisitos de software. Os analistas de requisitos devem estudar os *puzzles* e identificar as necessidades do cliente a partir das informações levantadas. O analista de requisitos deve ter em mente o objetivo definido para o projeto durante a realização dessa tarefa buscando identificar os requisitos que atendem a esse objetivo. Uma vez identificado um requisito o analista pode iniciar a tarefa de documentação deste requisito. Não é necessário que o analista de requisitos identifique todas as necessidades do cliente antes de iniciar a documentação dos requisitos. Ao contrário, é recomendável que a documentação seja iniciada logo após um requisito ser identificado. O negociador pode ser consultado sempre que houver dúvidas com relação

as informações disponíveis no *puzzle*. Caso o analista de requisitos perceba que existem informações faltando em um *puzzle* e que tais informações são relevantes para o levantamento de requisitos, ele deve solicitar mais informações ao negociador e colaboradores e o requisito deve participar, novamente, da próxima rodada do processo, e assim consecutivamente, até que todas as informações necessárias estejam disponíveis e que não haja mais dúvidas. Essa tarefa representa o objetivo principal da aplicação da engenharia de requisitos, que é a identificação dos requisitos de software. Ela foi incluída no modelo proposto por esse trabalho visando a torná-lo aderente a esse objetivo.

- **Documentar os requisitos identificados:** Conforme o objetivo definido, por Pressman (2011), para a tarefa de especificação de requisitos, após identificar os requisitos de software esses devem ser documentados para que a equipe de desenvolvimento de software possa implementá-los. Os analistas de requisitos, com auxílio da equipe de desenvolvimento, podem definir a forma que julgarem mais adequada. Após a criação do documento de requisitos esse deve ser validado pelo negociador na etapa do processo destinada a validações.

As tarefas definidas para esta etapa seguem as premissas definidas para a tarefa de especificação de Pressman (2011). Porém, ao invés de obter as informações sobre os requisitos a serem especificados de documentos criados por técnicas tradicionais de engenharia de requisitos, o analista deve usar os diagramas *puzzle*, gerados a partir da aplicação do processo proposto por este trabalho, como fonte de informações.

	Negociador	Analista de requisitos
Identificar os requisitos de software	C/I	R
Documentar os requisitos de software	C/I	R

Quadro 4: Matriz RACI da etapa de especificação
Elaboração própria

3.3.5 Validação

A validação dos requisitos é a última etapa antes que os requisitos possam ser enviados para a equipe de desenvolvimento. Essa etapa segue conforme a tarefa de validação definida por Pressman (2011) tendo apenas algumas adaptações para se adequar ao processo proposto por esse trabalho. As adaptações sugeridas buscam integrar os atores ao processo e delimitar atuação e responsabilidades dos participantes.

Nessa fase, o negociador deve verificar se os requisitos documentados pelo analista de requisitos atende às necessidades da empresa, se não estão ambíguos, se estão consistentes e se não existem omissões de informações no documento. Essa responsabilidade foi atribuída ao negociador, pois caso essa responsabilidade fique a cargo dos colaboradores o processo pode sofrer com divergências de opinião entre eles, dificultando a etapa de validação dos requisitos e assim tornando o projeto mais suscetível a atrasos.

Seguindo as premissas de Pressman (2011) para a validação dos requisitos, todas as correções necessárias neles devem ser feitas. Um requisito só poderá ser enviado para desenvolvimento após o negociador aceitá-lo como correto.

Essa etapa prevê apenas as tarefas de validação e correção dos requisitos e tem o negociador como principal ator do processo, uma vez que foi atribuída a ele a função de validar os requisitos especificados pelo analista de requisitos. Após todos os requisitos serem validados e o negociador ter dado sua posição final sobre eles uma nova rodada do processo pode ser iniciada, voltando para a etapa de levantamento das informações.

O negociador possui autonomia para tomar qualquer decisão sobre os requisitos, aceitando-os ou rejeitando-os. Caso o requisito seja aceito, ele irá entrar para a fila de desenvolvimento. Essa autonomia dada ao negociador visa a simplificar a etapa de validação dos requisitos.

Conforme comentado anteriormente as tarefas dessa etapa do processo são:

- **Validação dos requisitos:** O negociador deve validar todos os requisitos documentados na etapa de especificação. O negociador tem a

responsabilidade de verificar se os requisitos atendem às necessidades do cliente e se ajudam o software a atender o objetivo definido para o projeto. O negociador é a única pessoa com autonomia para aceitar um requisito e declará-lo pronto para desenvolvimento. Porém, ele pode consultar os colaboradores, em caso de dúvida. Caso o negociador identifique algum defeito no requisito esse deverá ser corrigido pelo analista de requisitos e então passará por nova validação. Entre as possibilidades de ação que um negociador pode tomar com relação a um requisito destacam-se: a aceitação do requisito (que o torna pronto para desenvolvimento), a rejeição definitiva do requisito (que o exclui de forma definitiva do processo), a rejeição por defeito (que obriga que esse requisito seja corrigido e posteriormente validado novamente) ou a rejeição por falta de informação (que deve ocorrer sempre que identificado que o requisito está incompleto por falta de informações na base de conhecimento do projeto para sua conclusão). Caso um requisito seja rejeitado por falta de informações, ele deve aguardar que uma nova rodada do processo seja iniciada para que informações complementares possam ser levantadas, a fim de que possa ser complementado. Essa tarefa foi especificada visando a atender ao objetivo da tarefa de validação definida por Pressman (2011).

- **Correção dos requisitos:** todos os defeitos nos requisitos identificados nessa etapa do processo devem ser corrigidos e nenhum requisito deve ser enviado à equipe de desenvolvimento se não for declarado como correto pelo negociador. Após as correções o requisito deve passar por uma nova validação do negociador e se nenhum outro defeito for encontrado esse requisito pode ser considerado como correto e então enviado para desenvolvimento. Essa tarefa foi definida com o objetivo de complementar a tarefa de validação no atendimento do objetivo definido por Pressman (2011).

As tarefas de validação seguem à premissa de Pressman (2011) para essa etapa da engenharia de requisitos em que os artefatos produzidos devem ser

validados e corrigidos. As tarefas definidas para essa etapa do processo de levantamento de requisitos foram adaptadas, buscando organizar melhor as atividades a serem executadas e definir as responsabilidades incumbidas aos papéis do processo.

	Negociador	Analista de requisitos
Validação dos requisitos	R	I
Correção dos requisitos	I	R

Quadro 5: Matriz RACI da etapa de validação
Elaboração própria

3.3.6 Ciclo de atuação contínua

É definido um ciclo contínuo, isso é, que ocorre durante todo o processo de levantamento de requisitos. Esse ciclo pode ser subdividido em três partes sendo uma focada nas negociações que ocorrem durante o projeto, uma focada na gestão dos *stakeholders* e uma focada na gestão de mudanças dos requisitos e informações. As tarefas definidas para o ciclo contínuo são baseadas nas tarefas de engenharia de requisitos do Pressman (2011), no conceito de inteligência estratégica antecipativa coletiva (BLANCK e JANISSEK-MUNIZ, 2014) e na ferramenta *puzzle* (ROUIBAH e OULD-ALI 2002).

Esse ciclo não faz parte do ciclo de levantamento de requisitos propriamente dito, pois sua atuação pode ocorrer em qualquer etapa do processo. As tarefas pertencentes a esse ciclo do processo podem ser consideradas comuns às demais etapas por estarem presentes durante todo o processo de levantamento de requisitos.

3.3.6.1 Negociação

A negociação é uma característica comum a todas as etapas do processo. Essa tarefa é baseada na tarefa de negociação definida por Pressman (2011), porém adaptada aos papéis definidos para o processo de engenharia de requisitos com inteligência coletiva proposto por este trabalho. Ela pode ocorrer, por exemplo, durante a etapa de levantamento de informações, quando houver divergências entre pontos de vista entre os colaboradores, durante a etapa de elaboração, envolvendo a forma como os sinais fracos são dispostos no *puzzle*, na etapa de especificação, podendo ser necessário negociar o escopo dos requisitos, e até mesmo na etapa de validação, em que pode ser necessário negociar com o cliente o nível de abrangência do requisito com relação às suas expectativas.

Esse item é, principalmente, de responsabilidade das pessoas que assumirem o papel de negociador. Os negociadores, além de atuar na negociação entre as partes, também devem atuar na gestão de conflitos que possam ocorrer. A gestão de conflitos também é muito importante do ponto de vista do projeto e ajuda a garantir a fluidez do processo.

A negociação não exige que seja produzido nenhum artefato, porém recomenda-se que em casos em que a negociação trate de um assunto delicado, o resultado dessa negociação seja formalizado por escrito e disponibilizado aos envolvidos.

As tarefas previstas para esse item do ciclo contínuo são:

- **Negociação:** a negociação, atividade equivalente à tarefa de mesmo nome proposta por Pressman (2011), pode ocorrer em qualquer etapa do processo de levantamento de requisitos, sempre com o objetivo de alcançar um acordo igualmente vantajoso para todas as partes envolvidas. Vários tipos de negociações podem estar presentes em um projeto de software. Alguns exemplos de negociações possíveis podem ser: negociação de escopo, de prazo e de nível de cooperação.

- **Gestão de conflitos:** durante o decorrer do projeto podem ocorrer conflitos ocasionados, por exemplo, por opiniões ou necessidades divergentes. Os negociadores devem interceder sempre que identificarem conflitos relacionados ao projeto, levando os envolvidos a encontrarem uma forma de resolução de conflitos. Um exemplo de conflito possível é quando colaboradores de um setor do cliente apontam uma necessidade para o software que entra em conflito com as necessidades de outro setor. A gestão de conflitos é um tipo de negociação. Porém, essa atividade ganhou um destaque maior no modelo proposto por este trabalho devido à grande quantidade de interações entre os participantes do processo.

3.3.6.2 Gestão de mudanças

A gestão de mudanças deve ocorrer sempre que haja alterações em qualquer informação do processo. A gestão de mudanças está presente desde a alteração de sinais fracos até requisitos documentados. Essa tarefa é derivada da tarefa de gestão de requisitos de Pressman (2011), porém adaptada para contemplar, também, os artefatos da ferramenta *puzzle*.

Uma mudança em um artefato pode impactar todos os demais artefatos, principalmente os derivados ou que possuam algum nível de interdependência do artefato alterado. É importante ter controle sobre as mudanças nos artefatos do processo e gerenciar os impactos envolvidos. A partir do controle das mudanças nos artefatos pode-se ter uma maior garantia com relação a qualidade dos requisitos gerados pelo processo de levantamento de requisitos.

O *puzzle* pode ser usado como base para identificar impactos ocasionados por alterações nos artefatos do processo. Em um *puzzle* podemos identificar quais artefatos são derivados do artefato alterado e as ligações entre os itens do *puzzle* também deixam explícito o impacto que uma alteração pode ter em outras informações interdependentes.

A gestão de mudanças não exige a criação de nenhum artefato específico e se beneficia dos já existentes. Porém, sugere-se que seja mantido o histórico das alterações ocorridas nos artefatos do processo para que possa ser consultado em caso de dúvidas.

Gerir mudanças é a única tarefa desse item do ciclo contínuo do processo de levantamento de requisitos e é inteiramente de responsabilidade dos analistas de requisitos.

3.3.6.3 Gestão de stakeholders e incentivo à participação no processo

As tarefas da gestão de *stakeholders* e incentivo à participação no processo de levantamento de requisitos de software são as únicas responsabilidades dos “animadores”. Essas tarefas foram definidas com base no conceito de inteligência antecipativa coletiva, que além de definir o papel de “animador” o define como o um facilitador do processo, com a missão de incentivar os participantes a colaborarem com a coleta dos sinais fracos.

Os “animadores” são as pessoas responsáveis por identificar os interessados no projeto e quais delas podem se tornar colaboradores do processo.

A segunda atribuição do “animador” consiste em incentivar os colaboradores a participarem do processo de levantamento de requisitos com inteligência coletiva e a contribuírem com seus conhecimentos.

Uma boa atuação do “animador” é muito importante para o levantamento de requisitos, pois esse processo é diretamente dependente do trabalho coletivo de levantamento das informações e compartilhamento de conhecimento. É dever do “animador” encontrar formas de motivar os colaboradores a participarem do processo de levantamento de requisitos.

As tarefas dessa etapa do ciclo são:

- **Gestão de *stakeholders*:** Essa atividade foi definida com base na tarefa de identificação dos participantes da inteligência estratégica antecipativa coletiva. Nessa tarefa os “animadores” devem fazer a gestão dos

stakeholders do projeto. Essa tarefa consiste em identificar todas as pessoas ou entidades que possuam algum interesse no projeto. Dentre os interessados identificados, o incentivador deve selecionar aqueles que podem atuar como colaboradores do projeto. Dentre os *stakeholders* identificados, o animador atuará principalmente com os selecionados para atuarem como colaboradores. Para os demais *stakeholders* basta manter um registro atualizado listando os interessados e quais são as suas relações com o projeto.

- **Incentivo à participação no processo:** Definida com base nas atribuições definidas para o “animador” da inteligência estratégica antecipativa coletiva, essa é a tarefa de maior importância dentre as atribuídas ao “animador”. As pessoas que assumem o papel de “animador” são responsáveis por estudar formas de incentivar os colaboradores a participarem ativamente do processo. Além de incentivar os colaboradores a participarem do processo, o “animador” deve se certificar que todos os colaboradores possuem acesso às ferramentas de coleta de informações e que não existem impedimentos para que possam desempenhar seu papel no processo.

	Negociador	Animador	Analista de requisitos
Negociação	R		
Gestão de conflitos	R		
Gestão de mudanças			R
Gestão de <i>stakeholders</i>		R	
Incentivo a participação no processo		R	

Quadro 6: Matriz RACI do ciclo contínuo
Elaboração própria

3.4 Artefatos

Os artefatos definidos para o modelo criado são saídas das tarefas do modelo de aplicação de inteligência coletiva e podem ser entradas de outras tarefas. Os artefatos foram definidos usando, como base, as tarefas da engenharia de requisitos de Pressman (2011), o conceito de inteligência estratégica antecipativa coletiva e a ferramenta *puzzle*.

3.4.1 Objetivo do projeto

Este é o primeiro artefato definido para o modelo apresentado. O artefato “objetivo do projeto” foi definido com base na descrição da tarefa de concepção apresentada por Pressman (2011).

Pressman (2011) afirma que a tarefa de concepção trata dos motivos que dão origem a um projeto de software. Visando a documentar esses motivos, para que possam ser usados como um guia ao processo de levantamento de requisitos, o artefato “objetivo do projeto” foi definido como uma descrição textual que informa o que é esperado do projeto de software.

Um bom objetivo deve conter o resultado esperado com a implantação do software, informar o alcance do software na organização e uma descrição de alto nível do sistema a ser desenvolvido.

Com a criação deste artefato, durante a execução da tarefa de definição do objetivo do projeto, espera-se que os motivos que dão origem ao projeto de software sejam formalizados e atendidos ao final do projeto de desenvolvimento de software.

3.4.2 Base de conhecimento

A base de conhecimento é um espaço, físico ou virtual, para armazenamento dos sinais fracos. Este artefato foi definido ao identificar-se a necessidade de armazenar os sinais fracos na aplicação da ferramenta *puzzle* e da inteligência estratégica antecipativa coletiva.

Sendo este um repositório dos sinais fracos, ele deve ser acessível a todos os participantes do processo de levantamento de requisitos. Os colaboradores devem poder incluir ou alterar qualquer sinal fraco disponibilizado para a aplicação do modelo de levantamento de requisitos com inteligência coletiva.

O objetivo da definição desse artefato é criar uma forma única de concentrar os sinais fracos coletados para um projeto de software e, assim, facilitar o processo de análise destas informações.

3.4.3 Temas

Os temas são artefatos incorporados da ferramenta *puzzle* e da inteligência estratégica antecipativa coletiva. Conforme a definição original pelas ferramentas de inteligência coletiva, os temas representam o assunto de aplicação do processo de levantamento de requisitos com inteligência coletiva.

Os temas são descrições textuais de assuntos a serem analisados para o levantamento de requisitos. Eles são definidos conforme relevância para o processo de levantamento de requisitos e podem ser considerados subdivisões lógicas, de alto nível, do projeto de software.

Como a aplicação do modelo de levantamento de requisitos proposto por este trabalho pode dar origem a mais de um *puzzle*, é possível que seja definido mais de um tema por projeto durante o processo de levantamento de requisitos com inteligência coletiva. A existência de múltiplos temas é possível, pois pode ser necessário subdividir o domínio do problema a ser resolvido, pelo desenvolvimento de software, em vários subdomínios, facilitando assim a aplicação do processo de

levantamento de requisitos e a coesão entre as informações agrupadas em um mesmo tema.

3.4.4 *Puzzle*

A ferramenta *puzzle* foi incorporada em sua totalidade como um artefato do modelo de levantamento de requisitos proposto por este trabalho. Cada tema definido para o levantamento de requisitos deve dar origem a um *puzzle*. As informações adquiridas após a aplicação dos *puzzles* são a origem dos requisitos de software, sendo o objetivo principal da incorporação deste artefato.

Um *puzzle* originado de um tema do software é uma representação gráfica dos sinais fracos desse tema combinados em informações e agrupados de forma lógica para criação de sentido. Um *puzzle* disponibiliza as informações agrupadas em um modelo visual para favorecer a compreensão por seu leitor. Nesse processo o *puzzle* é usado como uma forma de documentação dos requisitos do negócio, uma base central de conhecimento que pode ser utilizada pelo engenheiro de requisitos no momento de levantar e documentar os requisitos para desenvolvimento. Esse artefato é o resultado da aplicação da ferramenta *puzzle* em um contexto de levantamento de requisitos. Tal como o diagrama originalmente definido pela ferramenta *puzzle*, o diagrama *puzzle* gerado pela aplicação do processo definido por este trabalho possui informações importantes sobre um tema específico, separadas por categorias e levantadas a partir do trabalho colaborativo de coleção dos sinais fracos. Pode-se dizer, então, que o diagrama *puzzle* usado no processo de levantamento de requisitos de software com inteligência coletiva proposto por este trabalho é um modelo aplicado da ferramenta *puzzle*.

3.4.5 Documento de requisitos

Conforme Pressman (2011) e Sommerville (2011), o documento de requisitos é um artefato comumente utilizado na engenharia de requisitos. É utilizado para formalizar e documentar um requisito de software. Os requisitos identificados durante a execução dessa etapa do processo devem ser documentados. Eles podem seguir qualquer padrão desejado pela equipe de desenvolvimento e analista de requisitos. Porém, esse documento deve ser de fácil compreensão pelo negociador, pois ele será responsável por validar todos os requisitos relacionados ao tema pelo qual é responsável. O documento de requisitos deve servir como um contrato entre o cliente e a equipe de desenvolvimento, pois ele descreve, exatamente, o que vai ser implementado no software e representa a expectativa dos clientes.

3.4.6 Lista de aceitação dos requisitos

Este artefato foi definido como uma forma de organizar e classificar os requisitos na etapa de validação do processo de levantamento de requisitos. A lista de aceitação dos requisitos nada mais é que uma lista dos requisitos documentados na etapa de especificação, informando se esses requisitos foram aceitos ou rejeitados e o motivo de sua eventual rejeição. A partir dessa lista pode-se identificar a ação a ser tomada para cada requisito. Requisitos aceitos devem ser enviados para a equipe de desenvolvimento e requisitos rejeitados devem ser corrigidos ou removidos do processo conforme o motivo das rejeições. A lista de aceitação dos requisitos não é um artefato da engenharia de requisitos ou inteligência coletiva. Este artefato é de utilização opcional e é proposto como uma solução para facilitar a organização das tarefas da etapa de validação dos artefatos.

4 CONSIDERAÇÕES FINAIS

O modelo proposto por esse trabalho é baseado nas tarefas de engenharia de requisitos definidas por Pressman (2011), no conceito de inteligência estratégica antecipativa coletiva (BLANCK e JANISSEK-MUNIZ, 2014) e na ferramenta de criação de sentido coletivo *puzzle* (ROUIBAH e OULD-ALI 2002). As tarefas da engenharia de requisitos deram origem às etapas desse modelo e suas tarefas foram definidas conforme as tarefas da ferramenta *puzzle* e inteligência estratégica antecipativa coletiva.

Os papéis definidos para este modelo também foram baseados em estudos realizados sobre engenharia de requisitos e inteligência coletiva, sendo o negociador e o analista de requisitos baseados em papéis da engenharia de requisitos e o animador e o colaborador papéis baseados na ferramenta *puzzle*.

O modelo aqui proposto se trata de um trabalho inédito. Criado a partir da adaptação e união de conceitos e práticas de áreas distintas do conhecimento e com objetivo de apresentar uma metodologia alternativa para levantamento de requisitos de software.

As tarefas da engenharia de requisitos apresentadas por Pressman (2011) foram escolhidas como base para definição das etapas do modelo definido por esse trabalho por se mostrarem uma forma organizada para segregação das atividades da engenharia de requisitos. Para atingir o objetivo de propor uma metodologia de levantamento de requisitos de software com inteligência coletiva foram escolhidas as tarefas da ferramenta *puzzle* e da inteligência estratégica antecipativa coletiva para compor as tarefas específicas das etapas do modelo proposto.

A inteligência estratégica antecipativa coletiva e a ferramenta *puzzle* foram selecionadas devido as suas características de coleta, manipulação e utilização de dados em um tema específico, além da utilização em ambiente empresarial já testada. Ao contextualizar essas duas tecnologias em um processo de levantamento de requisitos de software, foi possível projetar uma forma organizada de incluir o trabalho coletivo dos *stakeholders* no levantamento de informações em um tema específico que, no caso, são os requisitos do software.

Percebeu-se que existe uma semelhança entre as fases da engenharia de requisitos e da ferramenta *puzzle*. A semelhança encontrada entre os conceitos estudados tornou possível a união desses em uma única e nova ferramenta de levantamento de requisitos.

O resultado esperado pela aplicação do modelo proposto por esse trabalho é uma maior aproximação entre os requisitos levantados e as necessidades do cliente e demais *stakeholders*, que poderão contar com um software melhor adaptado às suas tarefas e ao seu dia a dia. Acredita-se que com a aplicação desse modelo os analistas de requisitos terão à sua disposição uma fonte de dados preciosa sobre o negócio e o problema a ser resolvido. Os requisitos gerados a partir das informações levantadas por pessoas com vivência diária nos processos organizacionais devem-se mostrar mais próximos da necessidade do cliente e melhor adaptados às atividades operacionais a que se propõe suportar.

É importante salientar que pode haver a necessidade de adaptar o modelo proposto às diferentes realidades de projetos de software. Recomenda-se um estudo prévio no ambiente em que se pretende implantar o processo de levantamento de requisitos buscando descobrir limitações e impeditivos ao processo antes de sua utilização. Acredita-se que ambientes em que o trabalho em equipe e espírito colaborativo estejam mais presentes entre as pessoas sejam mais adequados à aplicação do processo de levantamento de requisitos com inteligência coletiva.

REFERÊNCIAS

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. Porto Alegre: Bookman, 2011.

SOMMERVILLE, Ian. **Software Engineering**. Boston: Pearson, 2011.

PMI. *Project Management Institute. Um Guia do Conhecimento em Gerenciamento de projetos (GUIA PMBOK®)*: 5 ed. Project Management Institute, Four Campus Boulevard, Newtown Square, Pennsylvania, USA, 2013. Tradução oficial de: "A Guide to the Project Management Body of Knowledge" (PMBOK® Guide)

PINTO, Augusto. **Gerenciamento de Projetos: Análise dos Fatores de Risco que Influenciam o Sucesso de Projetos de Sistemas de Informação**. 2002. 235f. Projeto de Dissertação, Universidade de São Paulo, São Paulo, 2002. Disponível em: <<http://docplayer.com.br/9425698-Gerenciamento-de-projetos-analise-dos-fatores-de-risco-que-influenciam-o-sucesso-de-projetos-de-sistemas-de-informacao.html>>. Acesso em: 3 maio 2016.

LUNARDI, Guilherme Lerch; DOLCI, Pietro Cunha. **Adoção de Tecnologia de Informação (TI) e seu Impacto no Desenho Organizacional: Um Estudo Realizado com Micro e Pequenas Empresas**. In ENCONTRO DA ANPAD, 30. 2006, Salvador. Anais... Salvador: Associação Nacional de Pós Graduação e Pesquisa em Administração, 2006. Disponível em <<http://www.anpad.org.br/enanpad/2006/dwn/enanpad2006-adic-1256.pdf>>. Acesso em: 3 maio 2016.

DALFOVO, Michael Samir; LANA, Rogério Adilson; e SILVEIRA, Amélia. **Métodos Quantitativos e Qualitativos: Um resgate teórico**. Disponível em <http://www.unisc.br/portal/upload/com_arquivo/metodos_quantitativos_e_qualitativos_um_resgate_teorico.pdf>. Acesso em: 7 maio 2016.

CARVALHO, Indymar Oliveira. **O Método PALAS (MCDA-C) no Levantamento dos Requisitos Construtivos de um Software de Controle de Uma biblioteca de Documentação Histórica**. Disponível em <http://tede.unifacs.br/tde_busca/processaPesquisa.php?listaDetalhes%5B%5D=258&processar=Processar>. Acesso em: 9 maio 2016.

FRANCETO, Simone. **Especificação e Implementação de uma Ferramenta para Elicitação de Requisitos de Software Baseada na Teoria da Atividade**. 2005. 133f. Projeto de Dissertação, Universidade Metodista de Piracicaba, Piracicaba, 2005. Disponível em: <<https://www.unimep.br/phpg/bibdig/pdfs/2006/VMEPWGALFNMD.pdf>>. Acesso em: 9 maio 2016.

PEREIRA, Ana Cele. **A Inteligência Coletiva: A Internet como Extensão do Conhecimento no Processo Ensino-Aprendizagem de Jovens do Ensino Médio.** Revista Lenpes-Pibid de Ciências Sociais – UEL On-line, Londrina, n.2, jul a dez. 2012. Disponível em: <<http://www.uel.br/revistas/lenpes-pibid/pages/arquivos/2%20Edicao/ANA%20CELE%20-%20Orient.%20angela.pdf>>. Acesso em: 15 maio 2016.

OMG. **OMG Unified Modeling Language™ (OMG UML):** Version 2.5. Disponível em: <www.omg.org/spec/UML/2.5/>. Acesso em: 21 junho 2016.

BELL,, Donald. **UML basics:** Part II: The activity diagram. Disponível em: <https://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep03/f_umlbasics_db.pdf>. Acesso em: 21 junho 2016.

BEMBEM, Angela; SANTOS, Plácida. **Inteligência coletiva:** Um olhar sobre a produção de Pierre Lévy. Revista Perspectivas em Ciência da Informação On-line, São Paulo, n. 4, out a dez. 2013. Disponível em: <<http://www.scielo.br/pdf/pci/v18n4/10.pdf>>. Acesso em: 3 julho 2016.

LOZZI, Rodolfo. **As árvores de conhecimentos.** Revista de Administração de Empresas On-line, São Paulo, n. 3, jul a set. 1996. Disponível em: <<http://www.scielo.br/pdf/rae/v36n3/a10v36n3.pdf>>. Acesso em: 3 julho 2016.

FERREIRA, Marcell; JANISSEK-MUNIZ, Raquel. **Aplicação de inteligência estratégica antecipativa e coletiva em empresa do setor varejista de calçados.** In Convibra, 29. 2013, Porto Alegre. Disponível em: <http://www.convibra.org/upload/paper/2013/29/2013_29_7977.pdf>. Acesso em: 9 julho 2016.

BLANCK, Mery; JANISSEK-MUNIZ, Raquel. **Inteligência estratégica antecipativa e crowdfunding:** Aplicação do método L.E.SCAning em empresa social de economia peer-to-peer (P2P). Revista da Administração da Universidade de São Paulo On-line, São Paulo, n. 1, jan a mar. 2014. Disponível em: <<http://www.scielo.br/pdf/rausp/v49n1/a15v49n1.pdf>>. Acesso em: 9 julho 2016.

ROUIBAH, Kamel; OULD-ALI, Samia. **PUZZLE:** A concept and prototype for linking business intelligence to business strategy. The Journal of Strategic Information Systems On-line, n. 11, june. 2002. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0963868702000057>>. Acesso em: 20 julho 2016.

XIBERRAS, Martine. **Internautas:** Inteligências coletivas na cibercultura. Revista Famecos – Mídia, Cultura e Tecnologia On-line, Porto Alegre, n. 17, set a dez. 2010. Disponível em: <<http://go.galegroup.com/ps/i.do?id=GALE%7CA306356836&v=2.1&u=capes&it=r&p>>

=AONE&sw=w&asid=d4a11dd2472db4a5595f04ceca274cbf>. Acesso em: 6 julho 2016.