

RECOVERY SOFTWARE REQUIREMENTS IN THE REVERSE ENGINEERING OF LEGACY SYSTEMS

Fernanda Vergani Santos Luiz (IPT - Instituto de Pesquisas Tecnológicas do Estado de SP, São Paulo, Brasil) – fernanda.vergani@gmail.com

Vagner Luiz Gava (IPT - Instituto de Pesquisas Tecnológicas do Estado de SP, São Paulo, Brasil) – vlgava@ipt.br

This article proposes a process for the recovery of the requirements of legacy systems using reverse engineering. The goal is to produce an updated and consistent specification for building a system to replace the legacy. The process uses three different sources of information for the requirements of the legacy system, which are subjected to technical analysis. It is used as a research methodology to Action Research. The process is built progressively in successive cycles of implementation and further refines that. Each cycle contributes in defining a sub process, which refines the result output in the previous cycle. The three sources used are: documents existing legacy system, the application of black box testing the functionality available in the interfaces Human-Human and Human-Machine and the execution traces produced by the legacy. These sources are exploited respectively in each cycle serving as stimuli of the sub processes.

Keywords: Software Reverse Engineering, Requirements Engineering, Legacy Systems, Recovery Requirements.

RECUPERAÇÃO DE REQUISITOS DE SOFTWARE NA ENGENHARIA REVERSA DE SISTEMAS LEGADOS

RESUMO

Este artigo propõe um processo para a recuperação dos requisitos de sistemas legados utilizando a engenharia reversa. O objetivo é produzir uma especificação atualizada e consistente para a construção de um sistema para substituir o legado. O processo utiliza três diferentes fontes de informação para obter os requisitos do sistema legado, que são submetidas às técnicas de análise. Utiliza-se como metodologia de pesquisa a Pesquisa-Ação. O processo é construído progressivamente em ciclos de execução sucessivos e complementares que o refina. Cada ciclo contribui na definição de um subprocesso, cuja saída refina o resultado obtido no ciclo anterior. As três fontes utilizadas no processo são: os documentos existentes do sistema legado, a aplicação de testes caixa preta nas funcionalidades disponíveis nas interfaces Homem-Homem e Homem-Máquina e os traços de execução produzidos pelo sistema. Estas fontes são exploradas respectivamente em cada ciclo servindo como estímulos dos subprocessos.

Palavras-chave: Engenharia Reversa de Software, Engenharia de Requisitos, Sistemas Legados, Recuperação de Requisitos.

1 Introdução

O ciclo de vida do software trata desde a necessidade inicial dos *stakeholders* (Todos os envolvidos no processo de especificação e uso do sistema informatizado), a concepção do sistema, a elicitação e análise dos requisitos de negócio, derivação dos requisitos de sistemas, projeto da solução, construção, verificação, validação, entrega até a sua manutenção. Este ciclo renova-se a cada nova demanda de mudança, seja ela corretiva ou evolutiva.

Esta renovação pode agregar novas funcionalidades ao sistema e algumas vezes, também, implicar em mudanças arquiteturais para dar suporte às funcionalidades mais complexas que por ventura possam surgir.

Segundo Pressman (2010), conforme consecutivas mudanças são realizadas em um software, é comum novos defeitos serem introduzidos, fazendo com que sua taxa de defeitos aumente progressivamente, sem que se consiga retornar ao nível ideal ao longo de seu ciclo de vida. Assim, constata-se que o software se deteriora em função das sucessivas manutenções sofridas.

Ao longo das manutenções do software, faz parte do processo de análise de viabilidade avaliar o custo sistêmico e o custo financeiro do projeto. Essa métrica permite, além de estimar o custo e esforço de um projeto, acompanhar sua condição ou tempo de vida útil. Dependendo da quantidade e da qualidade das manutenções efetuadas, pode-se chegar a uma situação na qual o custo envolvido acabe se tornando inviável para o retorno que tal manutenção produzirá.

Assim, deve-se tomar uma decisão estratégica sobre a continuidade do sistema, ou sobre a sua substituição.

Quando a empresa fornecedora do software identifica esta situação, pode tomar a iniciativa de construir um novo software que atenda aos mesmos requisitos de negócio do atual, mas fazendo uso de novas técnicas e linguagens de desenvolvimento que proporcionem uma arquitetura mais adequada para comportar evoluções com maior facilidade e, conseqüentemente, menores custos.

A partir desta decisão, faz-se necessário obter os requisitos do sistema legado detalhadamente, para dar início ao desenvolvimento do novo software. Ainda neste contexto, uma peculiaridade relevante para este artigo é que a iniciativa de construção de um novo software dá-se pelo fornecedor e não pelo cliente, motivada pelo elevado custo de manutenção e complexidade do código atual, de modo que a nova solução deve suportar todos os requisitos de software atualmente suportados nos diversos clientes onde o sistema opera.

Segundo Kaindl et. al (2010) , técnicas de levantamento de requisitos envolvem interações, estruturadas ou não, com as “fontes de conhecimento” das necessidades que o sistema deve resolver, seguidas dos respectivos registros. As fontes de conhecimento podem ser tanto as pessoas envolvidas (ativamente ou passivamente) nos processos de negócio, quanto os documentos que contenham tal conhecimento, ou ainda ambos.

Segundo Pohl (2010) e Sommerville e Kotonya (1998), existem diversas técnicas de levantamento e especificação de requisitos que podem ser utilizadas nas atividades do processo de engenharia de requisitos, e a escolha das técnicas

aplicadas variam de acordo com o projeto de desenvolvimento de software e a organização onde este projeto é executado.

Deste modo, os requisitos para a construção do novo sistema não são obtidos por meio de técnicas convencionais as quais envolvem entrevistas e reuniões com usuários ou requisitantes, como seria no processo de desenvolvimento de um software desde seu início.

2 Objetivo e questões da pesquisa

O principal objetivo deste trabalho, de acordo com a seção 1, é propor um processo para especificação de requisitos de software de sistemas legados, a partir da aplicação de técnicas de redocumentação, validação e verificação, e análise comportamental do sistema.

Admite-se como premissa a existência da documentação de requisitos do sistema legado, considerando-a como principal fonte de conhecimento para a aplicação do processo proposto, de modo que as técnicas de análise dinâmica e estática que serão aplicadas deverão validar e complementar os requisitos extraídos dos documentos analisados.

Assim, esta pesquisa permite responder à seguinte questão:

- Como aplicar técnicas de Engenharia de Software para a obtenção dos requisitos de um sistema legado por meio de um processo para a recuperação destes requisitos?

Este artigo está organizado da seguinte maneira: primeiramente são definidos os principais conceitos que embasam o processo proposto; em seguida estes conceitos são logicamente encadeados por meio do processo proposto e apresentado como resultado por meio de uma pesquisa-ação com base na teoria apresentada e, finalmente; são feitas recomendações para a continuação desta pesquisa.

3 Principais conceitos

Nesta seção são definidos os principais conceitos que embasam o processo proposto, inicialmente por meio da conceituação da engenharia reversa aplicada a sistemas legados em 3.1 e, posteriormente em 3.2, são apresentados os principais modelos, técnicas e diagramas para recuperação de requisitos: técnicas para recuperação de requisitos com base na análise comportamental do sistema legado, modelo de captura de requisitos de sistemas legados orientado ao processo de negócio, processo incremental de mineração sobre os rastros de execução (logs), diagrama para a reconstrução da arquitetura de software e técnica de análise de características do sistema legado para viabilizar a migração tecnológica.

3.1 A engenharia reversa aplicada a sistemas legados

A engenharia reversa, segundo Chikofsky e Cross (1990) é o processo de análise de um sistema que visa identificar os componentes do sistema e suas relações e criar representações do sistema em um nível superior de abstração. Além disso, afirmam que a engenharia reversa é um processo apenas de análise, no qual o software em causa não é modificado, já que a sua modificação, neste contexto, é endereçada pela reengenharia.

Chikofsky e Cross (1990) apresentam uma taxonomia sobre o termo “engenharia reversa” na engenharia de software, na qual colaboram com a adequada aplicação do termo, em diferentes contextos e níveis de utilização. A figura 1 é uma adaptação da taxonomia da relação entre os termos verificados no estudo realizado e os processos da engenharia de software, os quais podem ser entendidos como transformações entre, ou dentro, de níveis de abstração, em termos de fases do ciclo de vida do software.

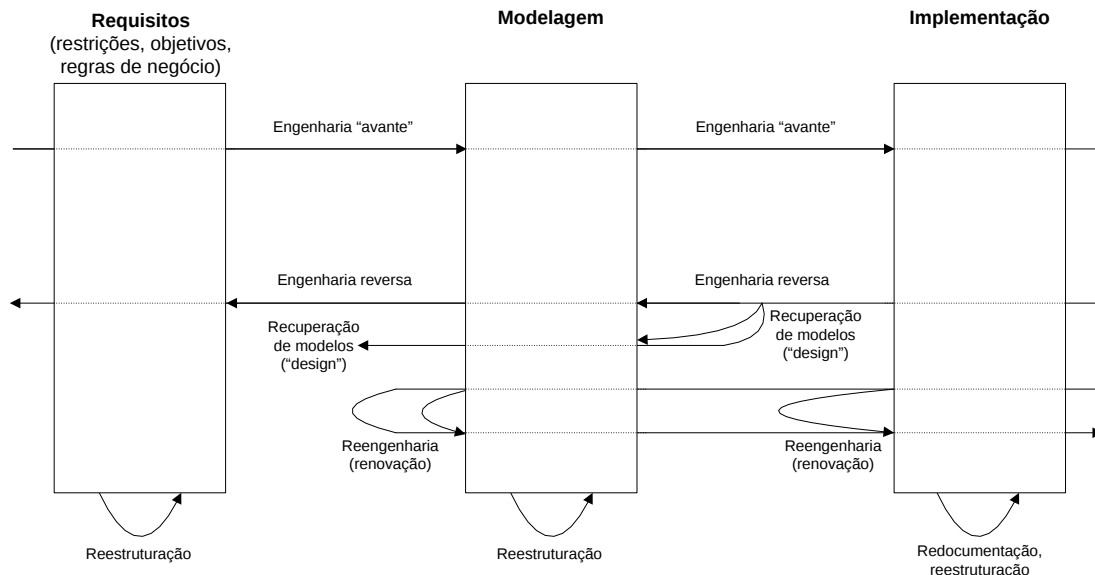


Figura 1: Relacionamento entre termos e processos
Fonte: Adaptado de Chikofsky e Cross (1990)

Segundo Sommerville (2010), a engenharia reversa de software auxilia a reconstrução dos sistemas legados para torná-los mais sustentáveis. Esta técnica pode envolver a redocumentação do sistema, a sua reestruturação, ou ainda a transformação da linguagem de programação de seu código fonte. No entanto, na reconstrução de sistemas legados por meio da engenharia reversa de software pura, a funcionalidade do sistema não é modificada, e normalmente a sua arquitetura também permanece a mesma.

Sommerville (2010) aponta as principais limitações da engenharia reversa de software como sendo a dificuldade de algumas mudanças arquiteturais dos sistemas legados como a distribuição de sistemas centralizados, ou ainda a conversão de linguagens de programação muito antigas, procedimentais, em linguagens de programação orientadas a objetos, como Java ou C++. Além disso, ele ressalta que limitações inerentes ao sistema também são mantidas, pois a funcionalidade do software permanece inalterada.

Caso estas limitações sejam indesejadas pela organização, a única alternativa possível para a evolução do sistema legado e a consequente manutenção do negócio suportado por ele é a sua substituição por novos sistemas. De qualquer forma, mantêm-se a necessidade de se conhecer os requisitos de software do sistema legado, para que estes sejam submetidos aos novos sistemas que o irão substituir, o que continua a exigir a aplicação da engenharia reversa, no mínimo, para recuperação dos requisitos de software.

3.2 Técnicas para recuperação de requisitos

A reconstrução de sistemas legados é um assunto que tem sido bastante abordado por pesquisadores nos últimos anos, conforme se constata nos trabalhos de Ducasse e Pollet (2009), Kalsing et. al. (2010), Liu et. al. (1999), Millham (2010) e Rub e Issa (2009).

A decisão de aplicação da engenharia reversa é uma alternativa interessante à reconstrução de um sistema legado, que minimiza os riscos envolvidos nesta última e implica, em geral, em um menor custo para a organização. No entanto, esta técnica nem sempre é viável, dependendo das evoluções tecnológicas que se pretenda fazer ao sistema legado, como, por exemplo, mudança de plataforma, conforme constata Sommerville (2010).

Além disso, outros fatores externos, como institucionais e políticos, podem conduzir a organização a decidir por seguir com a reconstrução do sistema legado, descartando o uso da engenharia reversa pura.

No entanto, para se reconstruir um sistema legado, é necessário conhecer todos os requisitos de software que foram implementados ao longo do ciclo de vida do sistema.

Segundo Sommerville (2010), sistemas legados não contam, em geral, com documentação suficientemente completa e atualizada para sua reconstrução. Além disso, Liu et. al. (1999) afirmam que quando há documentação, existe uma grande dificuldade em se entender os requisitos originais do sistema devido à descontinuidade da experiência dos envolvidos no processo de desenvolvimento ao longo do ciclo de vida do sistema.

Alguns processos e técnicas têm sido desenvolvidos e propostos pela comunidade científica para a captura, extração ou recuperação do conhecimento e dos requisitos de software de sistemas legados. Estes trabalhos ressaltam diferentes problemáticas que são utilizadas como diretrizes para sua elaboração.

Em Liu et. al. (1999), são aplicadas técnicas de recuperação de requisitos com base na análise comportamental do sistema legado. Estas técnicas ressaltam a recuperação dos requisitos por meio da modelagem dinâmica do sistema e desconsideram o uso das especificações de requisitos existentes.

No trabalho proposto neste artigo, as especificações de requisitos existentes não serão desconsideradas, por outro lado, será complementada pelas técnicas de captura do comportamento do sistema propostas no processo segundo Liu et. al. (1999).

A Figura 2 resume o processo proposto por esta pesquisa para a recuperação de requisitos. Este processo divide-se em três partes:

- Na primeira, constrói-se um modelo de processo semântico que produzem cenários a partir da análise de uma série de fontes de conhecimento, entre elas, o estudo do ambiente de TI no qual o sistema legado se encontra, a análise de contexto e semântica, assim como a observação das interações Homem-Máquina. A partir da análise dos cenários produzidos na primeira parte;
- Na segunda parte é feita a modelagem dinâmica, complementada pela

análise de normas e informações adicionais do ambiente de TI e

- Finalmente, a partir do resultado produzido na segunda parte, é feita a derivação de requisitos, chegando-se aos requisitos do sistema legado.

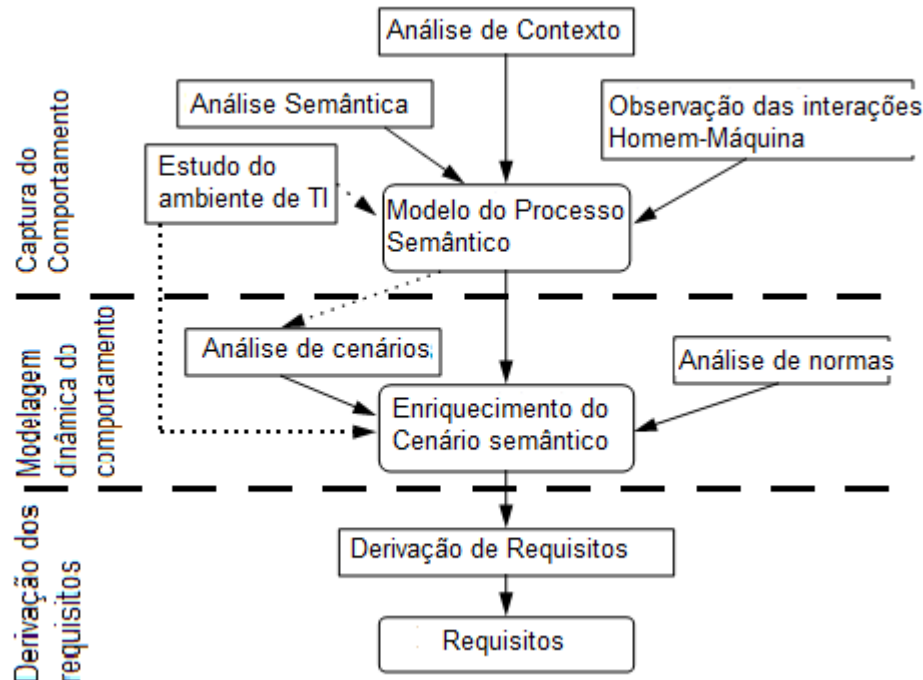


Figura 2: Processo de recuperação de requisitos AMBOLS
Fonte: Adaptado de Liu et. al. (1999)

Por outro lado, Rub e Issa (2009) apresentam um modelo de captura de requisitos de sistemas legados, orientado ao processo de negócio para a recuperação dos principais requisitos do software. Esta pesquisa é dividida em duas fases:

- A primeira trabalha na recuperação dos requisitos do sistema legado e na sua remodelagem por meio das técnicas RAD (*Role Activity Diagram*) e, na sequência, a UML com diagramas de Casos de Uso e;
- A segunda fase utiliza o resultado da primeira fase como entrada para a reconstrução do sistema legado por meio do processo de desenvolvimento convencional (*forward engineering*).

No entanto, ao analisar a primeira fase da pesquisa de Rub e Issa (2009) observa-se que no processo proposto para a recuperação de requisitos não é definida nem aprofundada a exploração das fontes de conhecimento utilizadas, atendo-se apenas à remodelagem destes requisitos por meio da técnica RAD, a partir do conhecimento do sistema legado já consolidado.

A pesquisa realizada neste artigo procura aprofundar-se nesta atividade, já que para a remodelagem do software tratada nesta pesquisa é fundamental a extração prévia do conhecimento dos requisitos do sistema legado, independente da técnica de remodelagem a ser utilizada no processo de recuperação de requisitos.

Kalsing et. al. (2010) propõem um processo incremental de mineração sobre os rastros de execução (logs) para a extração de conhecimento de sistemas legados que considera tanto a análise estática do sistema, como também a análise comportamental, ou seja, dinâmica. O diagrama que resume este processo é apresentado na Figura 3.

Um dos aspectos da pesquisa de Kalsing et. al. (2010) é o fato de considerar a modificação (ou manipulação) do código fonte do sistema legado como parte do processo, conforme se observa na segunda atividade do processo apresentado na Figura 3, com o propósito de gerar registros do comportamento do software aos estímulos sofridos por este e desta forma obter informações detalhadas para análise.

Mas, nos casos em que o código do sistema legado não pode ser modificado ou manipulado, como colocado na premissa deste trabalho, a aplicação desta técnica pode ser inviável, comprometendo a aplicação do processo proposto neste trabalho.

Por outro lado, esta pesquisa mostra a relevância desta fonte de conhecimento do sistema legado, quando considera como premissa a identificação das regras de negócio e construção dos casos de uso, respectivamente a primeira e a terceira atividades do processo apresentado na Figura 3.

Por isso será considerada neste artigo a análise dos registros de execução que já são inerentemente gerados pelo sistema legado, do modo como são implementados, sem a necessidade de modificações no seu código fonte.

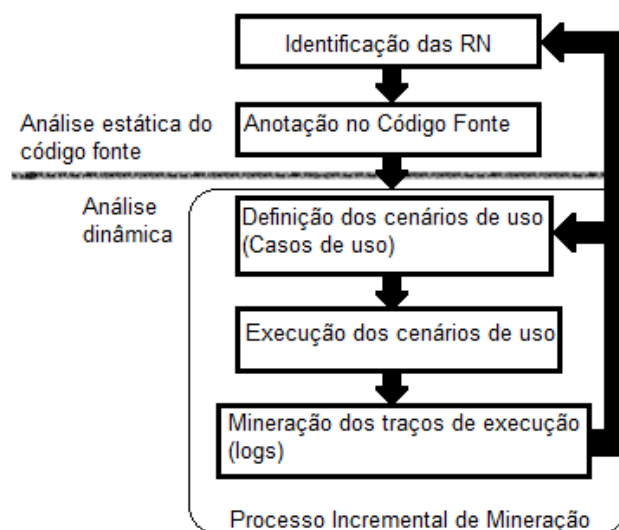


Figura 3: Processo Incremental de Mineração
Fonte: Adaptado de Kalsing et al. (2010)

As pesquisas estudadas até agora abordam a análise comportamental do sistema legado, no entanto, não consideram as documentações de especificação existentes destes sistemas como fonte relevante de conhecimento para a recuperação dos requisitos de software.

Ducasse e Pollet (2009) apresentam um diagrama produzido a partir da taxonomia de outros trabalhos que tratam a reconstrução da arquitetura de softwares. O diagrama produzido é intitulado por Taxonomia orientada aos processos de “SAR” (*Software Architecture Reconstruction*), conforme apresentado na Figura 4.

Este diagrama aborda cinco aspectos relacionados à reconstrução da arquitetura de software, entre elas, as metas (objetivos), os processos, as técnicas, as entradas utilizadas e as saídas produzidas pelos processos.

Em Ducasse e Pollet (2009) ressalta-se a necessidade de recuperação do conhecimento contido no sistema legado por meio da aplicação de técnicas, manuais ou automatizadas, para se alcançar um objetivo, que pode ser a redocumentação, o reuso, a análise, a evolução ou a gestão dos sistemas legados.

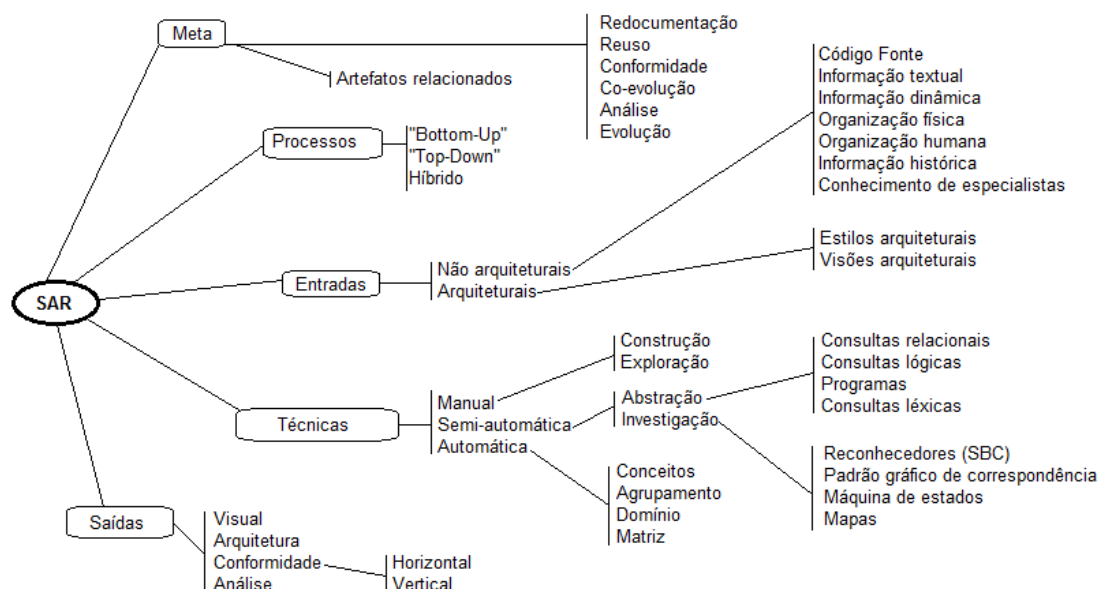


Figura 4: Taxonomia orientada aos processos de “SAR”
Fonte: Adaptado de Ducasse e Pollet (2009)

Já Millham (2010) propõe a utilização da técnica de análise de características do sistema legado para viabilizar a migração tecnológica, ou seja, a migração de uma linguagem procedimental (no caso, o COBOL) para uma linguagem orientada a objetos (utilizando o padrão de orientação a serviços). Apesar do objetivo desta pesquisa não ser propriamente a reconstrução do sistema legado, e sim a reengenharia aplicada à conversão arquitetural, a necessidade de identificação das funcionalidades do sistema legado é evidenciada e tratada como o primeiro passo para se alcançar este objetivo.

Na pesquisa de Millham (2010) é utilizada a técnica de análise de domínio, juntamente com uma ferramenta (TAGDUR), que auxilia na análise de dados e controle de dependências de código fonte para a identificação e agrupamento das funcionalidades do software. As técnicas aplicadas neste trabalho não atuam na

redocumentação dos requisitos do sistema legado, mas apenas na sua remodelagem parcial.

As pesquisas recentes, apresentadas nesta seção, ou seja, Liu et. al. (1999), Rub e Issa (2009), Kalsing et. al. (2010), Ducasse e Pollet (2009), Millham (2010) apoiam-se fundamentalmente na análise de código fonte do sistema legado, ignorando o conhecimento registrado em documentações disponíveis, já que seu objetivo não é propriamente a recuperação dos requisitos do sistema legado, e sim sua remodelagem e transformação.

Desta forma, aprofundam-se na identificação do domínio e principais funcionalidades, ou seja, restringem-se à análise estática e dinâmica do sistema legado sem aprofundarem-se no contexto no qual o sistema está inserido e as necessidades que deram origem aos requisitos de software implementados no projeto do sistema legado.

Pode-se observar também, nas pesquisas citadas, que em qualquer processo proposto para extração de conhecimento de sistemas legados, é necessário inicialmente identificar os atores (humanos ou sistemas externos) envolvidos nos processos de negócio, assim como os cenários de negócio nos quais estes processos são executados; no entanto tais pesquisas não se aprofundam neste propósito.

A identificação dos atores envolvidos nos cenários de negócio não é possível ser feita por meio da análise de código, uma vez que este não reconhece os estimuladores dos processos de negócio nele implementados. Desta forma, conforme constatado por Kalsing et. al. (2010), torna-se necessário recorrer a processos de análise não baseados em código fonte puramente, como entrevistas e reuniões com usuários, desenvolvedores, analistas de negócio e análise de documentações, por exemplo.

De acordo com Śmiałek (2007), o conhecimento adquirido em projetos passados, por meio dos requisitos especificados, pode ser reutilizado em projetos futuros semelhantes nos quais os mesmos conhecimentos são novamente demandados.

Ainda de acordo com este autor, a principal dificuldade de reutilização deste conhecimento está na forma como os requisitos são especificados na maioria dos projetos, ou seja, em linguagem natural, muitas vezes sem nenhuma estrutura.

Śmiałek (2007) conclui que, para que o reuso de requisitos seja eficiente, é necessário que as especificações de requisitos sejam bem estruturadas, apoiadas em modelos e linguagens adequadas para que os requisitos possam ser comparados e extraídos do repositório de forma sistemática (também deve haver um histórico de requisitos similares).

4 Metodologia de pesquisa

Esta pesquisa utiliza o método de pesquisa-ação (PA) que corresponde a um tipo de pesquisa social com base empírica que é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo e no qual os pesquisadores representativos da situação ou do problema estão envolvidos de modo cooperativo ou participativo (THIOLLENT, 2008:10). Coughlan e Coughlan (2002:230) expandem este conceito agregando uma série de características:

- O pesquisador toma uma ação (não é mero observador);
- A pesquisa-ação envolve dois objetivos: solucionar um problema e contribuir para a ciência;
- A pesquisa ação é interativa (cooperação e interatividade entre os envolvidos);
- A pesquisa-ação objetiva desenvolver um entendimento holístico;
- É fundamentalmente relacionada à mudança;
- A pesquisa ação pode incluir todos os tipos de coleta de dados (técnicas quantitativas e qualitativas);
- Requer um pré-entendimento do ambiente organizacional, das condições da estrutura e dinâmica das operações, etc.;
- Deve ser conduzida em tempo real e
- Requer critérios próprios de qualidade para sua avaliação.

Estas características devem ser consideradas desde o momento da concepção da pesquisa, de modo que a pesquisa-ação compreenda três fases principais: uma preliminar, um ciclo de condução e uma meta fase.

Como pode ser visto na Figura 5, o ciclo de condução da pesquisa propriamente compreende seis passos principais, sendo que a meta fase está presente em cada um destes seis passos. Durante a apresentação dos resultados, cada um dos passos será contextualizado dentro das respectivas fases.

A primeira fase (fase preliminar) corresponde a um entendimento sobre o contexto onde a pesquisa será realizada, bem como o propósito da condução do trabalho. Essa fase envolve ainda o estabelecimento das justificativas para a ação requerida, além das justificativas para a pesquisa.

A segunda fase inicia com a coleta de dados (diagnóstico e/ou dados coletados quando a pesquisa encontra-se em regime), feedback (para os envolvidos), análise (com os envolvidos), planejamento da ação (definição da intervenção a ser feita), implementação da ação (colocar em prática o que foi planejado) e avaliação (verificar se os resultados da implementação surtiram ou não efeitos desejados). Retornando para uma nova coleta de dados (se necessário), encerrando o ciclo (primeira iteração).

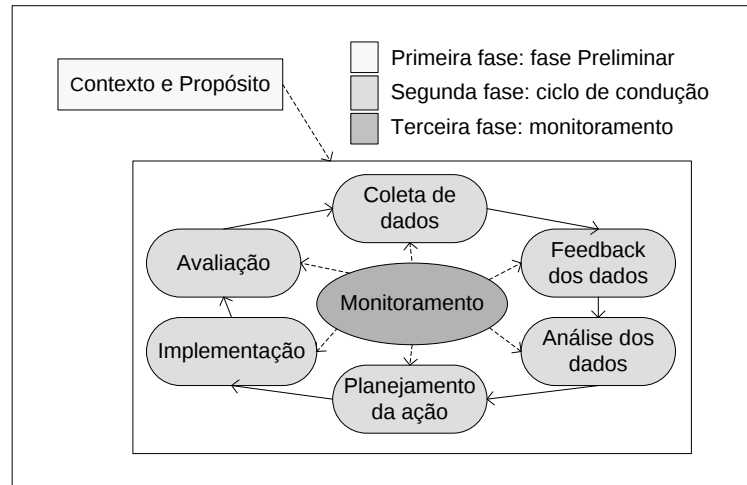


Figura 5 – Ciclos da Pesquisa-ação
Fonte: baseado em Coughlan e Coughlan (2002)

Esses ciclos são constantes e sequenciais, sendo contínuos pelo período em que forem necessários, podendo existir um ciclo mais abrangente (para a pesquisa como um todo) e ciclos menores para partes específicas do trabalho.

A terceira fase compreende uma verificação de cada um dos seis passos anteriores, de modo a verificar qual é o aprendizado gerado na condução da pesquisa-ação, de modo que o pesquisador deva estar interessado não somente na operação do projeto, mas também no monitoramento do processo de aprendizagem, que levará à contribuição teórica deste tipo de desenvolvimento empírico (MIGUEL, 2005:15).

5 Planejamento da execução do processo em função da Pesquisa-Ação

Este trabalho será executado em três diferentes ciclos de refinamento, que apresenta uma simplificação do ciclo sugerido por Coughlan e Coughlan (2002), no entanto, todos se encaixam ao modelo apresentado na Figura 5.

Cada ciclo se concentrará em um tipo de fonte de requisitos, ou seja, o primeiro ciclo concentra as atividades relacionadas à extração de requisitos dos documentos existentes do sistema legado, o segundo concentra-se na execução e análise de resultados dos testes Caixa Preta, enquanto que o último ciclo concentra-se na análise dos traços de execução (logs) do sistema legado. Esta estrutura constituída por três diferentes ciclos é apresentada na Figura 6.

Com o propósito de testar o processo, este será executado em um domínio de negócio restrito, embora o sistema legado apresente um conjunto de funcionalidades muito maior, pertencentes a diferentes domínios de negócio dentro dos ditos sistemas de suporte ao negócio (*BSSs – Business Support Systems*) no segmento das telecomunicações.

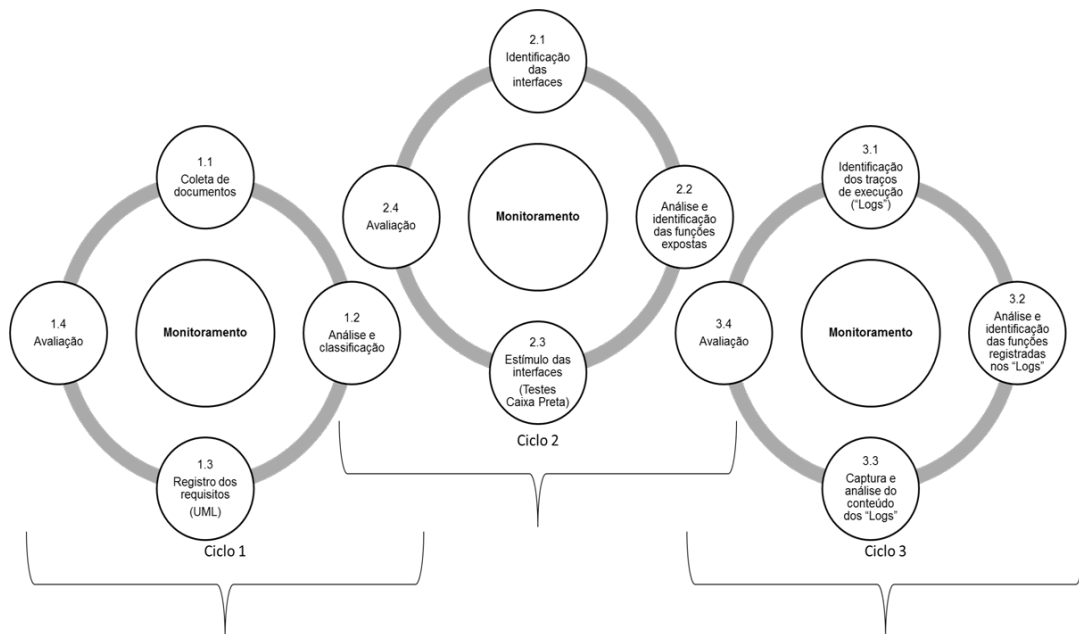


Figura 6: Iteração dos ciclos da PA aplicadas ao trabalho
Fonte: Autores

6 Resultados

Após a execução dos ciclos da Pesquisa-Ação, conforme descrito na seção 4, o processo resultante pode ser visto na Figura 7.

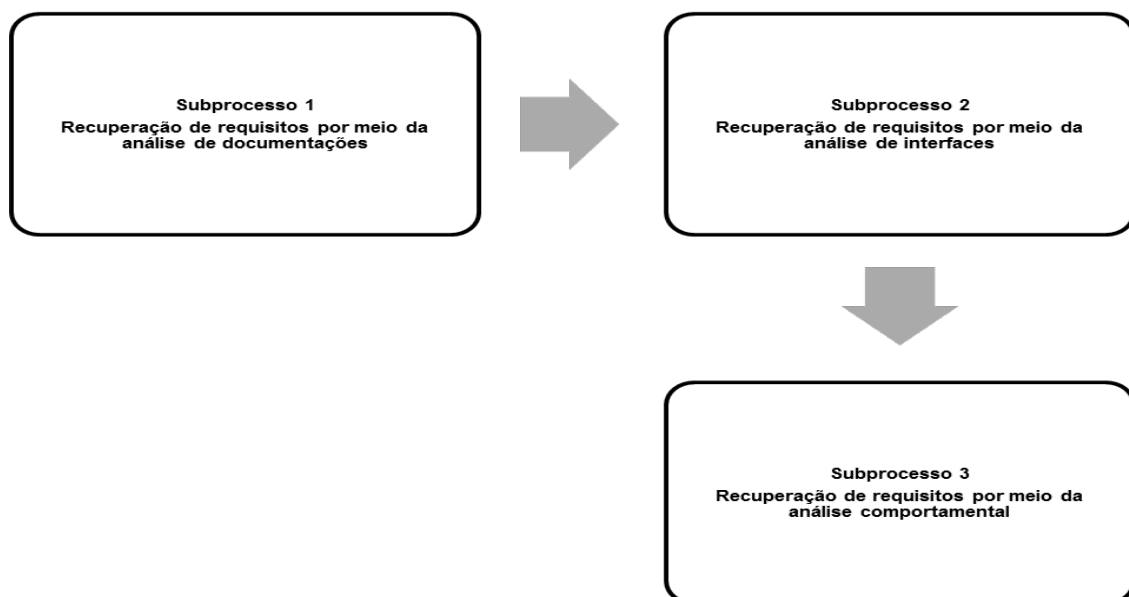


Figura 7: Processo proposto de recuperação de requisitos do sistema legado
Fonte: Autores

De modo que cada subprocesso pode ser detalhado em suas respectivas atividades, conforme pode ser visto na Figura 8.

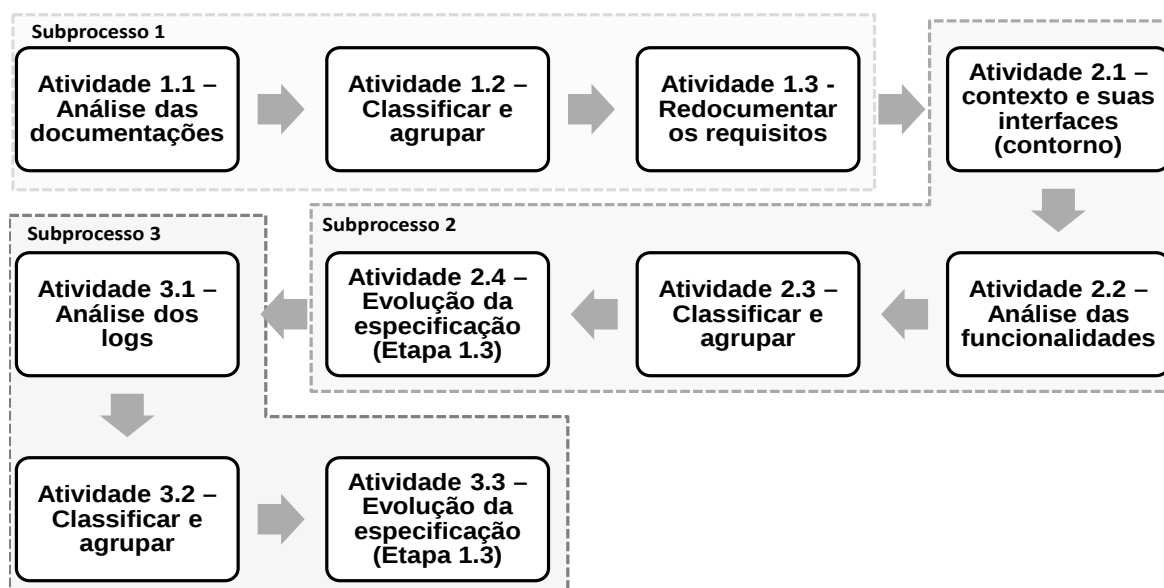


Figura 8: Atividades dos subprocessos derivados do processo inicial
Fonte: Autores

Nas seções seguintes, cada subprocesso e suas respectivas atividades e procedimentos serão descritos com mais detalhes. A nomenclatura utilizada neste trabalho para os diferentes níveis de agrupamento das atividades propostas, ou seja, processo, subprocesso, atividade e procedimento, baseia-se na estrutura proposta por CRUZ, T. (2004).

6.1 Subprocesso 1- Recuperação de requisitos por meio da análise de documentações

A figura 9 apresenta os procedimentos envolvidos nas atividades do subprocesso 1 deste artigo, considerando-se as entradas, saídas e mecanismos envolvidos em cada atividade.

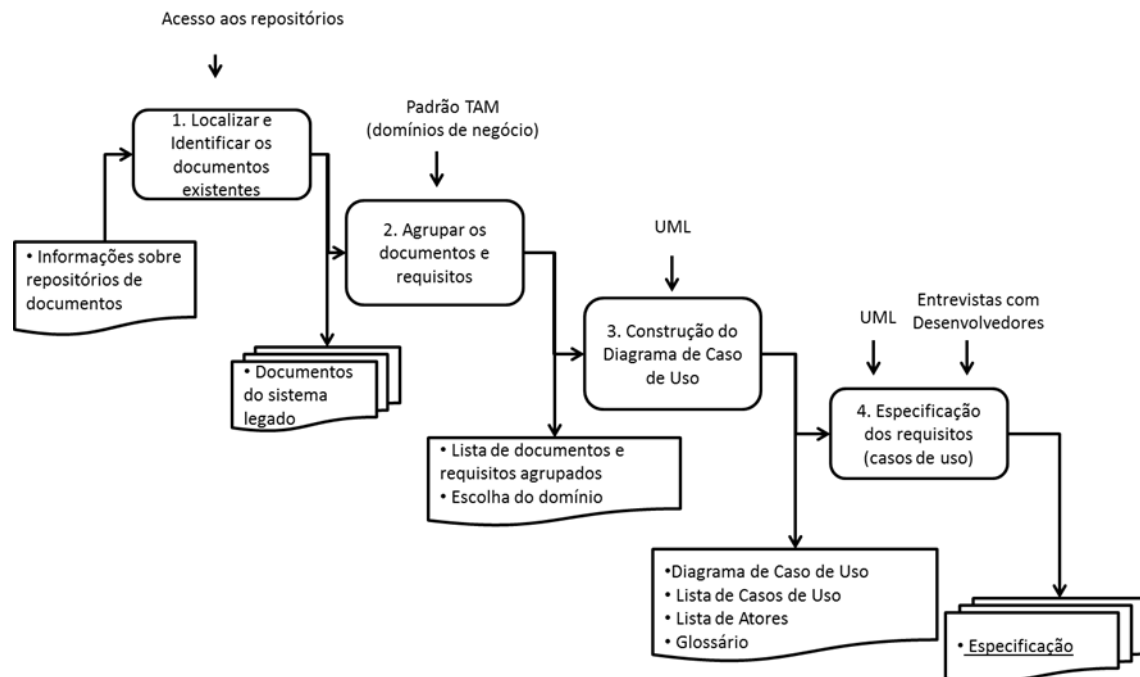


Figura 9: Procedimentos do subprocesso de recuperação de requisitos por meio da análise de documentações (subprocesso 1)

Fonte: Autores

6.1.1 Atividade 1.1 – Análise das documentações disponíveis para obtenção dos requisitos funcionais e não funcionais

Nesta atividade, todos os documentos de requisitos e arquitetura disponíveis devem ser analisados a fim de obter os requisitos funcionais e não funcionais do software.

Recomenda-se, sempre que necessário, recorrer aos desenvolvedores e especialistas do sistema para uma análise manual do código fonte com o objetivo de esclarecimento de requisitos incompletos ou ambíguos obtidos dos documentos analisados.

A seguir apresenta-se o procedimento detalhado desta atividade.

6.1.1.1 Procedimento 1: Identificação e localização dos documentos existentes

Descrição: Para se iniciar o subprocesso 1, a primeira atividade é identificar o repositório (ou os repositórios) onde os documentos existentes do sistema legado são armazenados e localizá-los.

Entrada: Informações (máquinas, diretórios, usuário e senha para acesso, padrões de nomenclatura dos documentos, padrões de estrutura dos documentos) sobre os repositórios onde se localizam as documentações existentes do sistema legado.

Saída: Lista dos documentos existentes do sistema legado (especificações de requisitos, especificações de interfaces, visões arquiteturais, projeto de desenvolvimento).

6.1.2 Atividade 1.2 – Classificando e agrupando os documentos e requisitos levantados

Os documentos devem ser agrupados de acordo com os domínios de negócio que tratam, assim como deverão ser organizados os requisitos extraídos deles.

Para cada domínio de negócio deve ser construída uma ficha de requisitos para fins de organização e localização da informação.

A seguir apresenta-se o procedimento detalhado desta atividade (ver figura 9, 2º bloco).

6.1.2.1 Procedimento 1: Agrupamento dos documentos e requisitos por domínio de negócio

Descrição: Após a localização dos documentos, os analistas devem realizar uma primeira leitura com o objetivo de agrupá-los de acordo com o domínio de negócio que tratam, concentrando-se desta forma as fontes de conhecimento a serem analisadas durante a pesquisa-ação, que será realizada sobre um único domínio com o objetivo de reduzir o escopo do trabalho.

Entrada: Documentos existentes do sistema legado

Saída: Documentos e requisitos classificados de acordo com o domínio de negócio escolhido (restrição de escopo do trabalho).

6.1.3 Atividade 1.3 – Redocumentação dos requisitos levantados

Esta atividade é constituída pela produção do diagrama de Casos de Uso, assim como a lista dos casos de uso com uma breve descrição.

A seguir apresentam-se os procedimentos detalhados desta atividade (ver figura 9, 3º e 4º blocos).

6.1.3.1 Procedimento 1: Construção do Diagrama de Casos de Uso

Descrição: Após o agrupamento dos documentos por domínio de negócio, os analistas devem realizar uma segunda leitura para o grupo que especifica os requisitos de software do domínio escolhido. Esta leitura deve ter por objetivo a construção do diagrama de Casos de Uso, e o início da construção da lista de Casos de Uso com o preenchimento dos campos indicados a seguir e a lista de Atores.

Caso de Uso (ID)
Nome do Caso de Uso
Documento Referência (ID)

Entrada: Grupo de documentos e requisitos do domínio de negócio que será alvo do trabalho desenvolvido nesta pesquisa

Saída: Diagrama de Casos de Uso, Lista de Casos de Uso (semi-preenchida) e Lista de Atores.

6.1.3.2 Procedimento 2: Especificação dos requisitos por meio de casos de uso

Descrição: Uma última leitura mais aprofundada dos documentos do domínio escolhido deve ser realizada com o objetivo de especificar cada Caso de Uso

levantado. Nesta última atividade os analistas deverão recorrer à consulta dos desenvolvedores especialistas para esclarecimento de ambiguidades ou dúvidas oriundas da análise das documentações.

Entrada: Grupo de documentos e requisitos do domínio de negócio que será alvo do trabalho desenvolvido nesta pesquisa, diagrama de casos de uso construído na atividade anterior e listas dos casos de uso e dos atores.

Saída: A lista de Casos de Uso deve ser completada com o preenchimento do campo “Descrição” e os analistas devem criar os requisitos de software derivados da análise inicial, referenciando a nova fonte de conhecimento, ou seja, a ficha criada a partir do diagrama e das listas de casos de uso e atores produzidos durante este ciclo.

6.2 Subprocesso 2 - Recuperação de requisitos por meio das interfaces

Nesta seção são detalhados os procedimentos que constituem o subprocesso 2, apresentado na Figura 10.

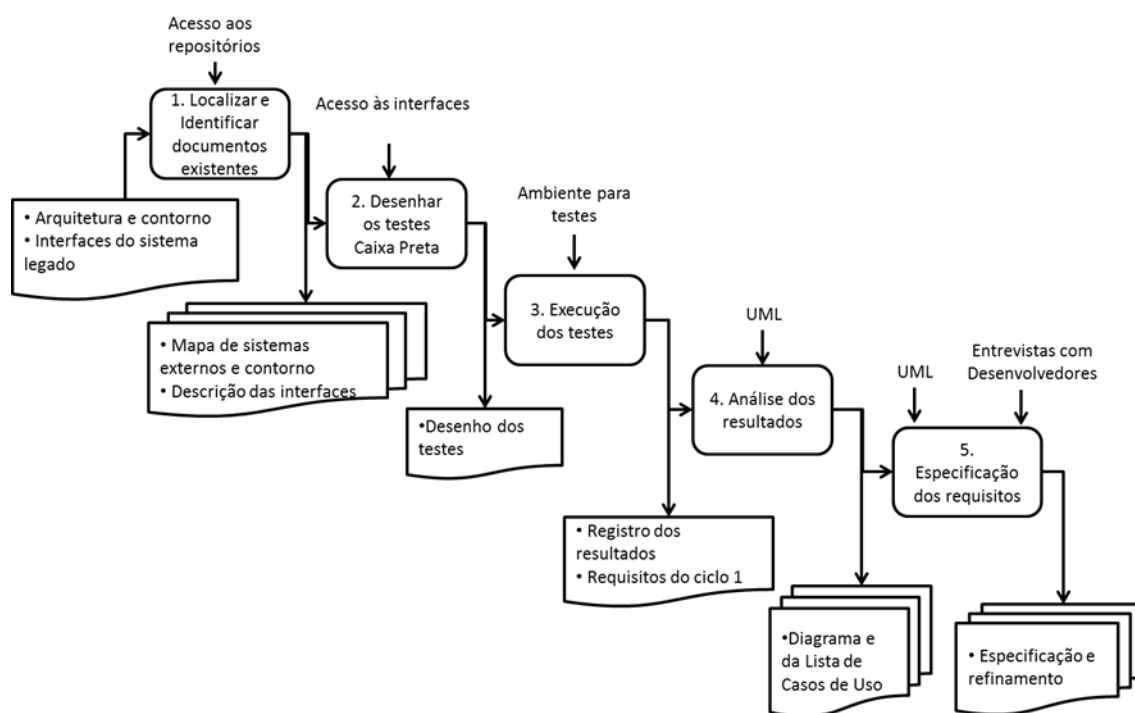


Figura 10: Procedimentos do subprocesso de recuperação de requisitos por meio da análise de interfaces

Fonte: Autores

6.2.1 Atividade 2.1 – Identificando o contexto de utilização do sistema e suas interfaces (sistemas externos e usuários)

Nesta atividade deve ser realizada uma análise de contexto para a identificação do ambiente em que o sistema legado está enquadrado, listando os sistemas externos com os quais interage, assim como as interfaces com usuários (GUI – *Graphic User Interfaces*) que o sistema legado fornece.

Esta atividade é concluída com a produção de um mapa de sistemas externos, assim como a lista das interfaces do sistema legado com estes sistemas, com uma breve descrição.

A seguir, apresenta-se o procedimento detalhado desta atividade.

6.2.1.1 Procedimento 1: Localizar e identificar os documentos existentes

Descrição: Para se iniciar o subprocesso 2, a primeira atividade é identificar o repositório (ou os repositórios) onde os documentos existentes relacionados à arquitetura, contorno e interfaces do sistema legado são armazenados e localizá-los.

Entrada: Informações (máquinas, diretórios, usuário e senha para acesso, padrões de nomenclatura dos documentos, padrões de estrutura dos documentos) sobre os repositórios onde se localizam as documentações existentes do sistema legado.

Saída: Lista dos documentos existentes do sistema legado (especificações de requisitos, especificações de interfaces, visões arquiteturais, projeto de desenvolvimento).

6.2.2 Atividade 2.2 – Análise das funcionalidades disponíveis nas interfaces

A partir desta lista de interfaces, uma segunda atividade deve ser realizada, aprofundando-se nos processos visíveis e disponíveis em cada uma das interfaces identificadas.

Além destes processos, deve-se observar a existência de processos internos, como relógios que desencadeiam outros processos no sistema que gerem saídas funcionais, como a geração de relatórios, alterações nas informações armazenadas pelo sistema ou ainda estímulos a outros sistemas com os quais interage.

A seguir, apresentam-se os procedimentos detalhados desta atividade.

6.2.2.1 Procedimento 1: Desenhar e Planejar os testes Caixa Preta (análise funcional das interfaces)

Descrição: A preparação dos testes deve ser feita com a produção de uma ficha que indique as condições em que cada teste foi executado, assim como as manipulações prévias de dados para a viabilização das condições indicadas.

A partir as informações coletadas na atividade anterior, os testes Caixa Preta serão planejados, tomando como base a seguinte regra:

- Serão realizados testes para os casos de uso já identificados e especificados nas atividades anteriores que não estiverem suficientemente completos;
- Serão realizados testes para os novos casos de uso identificados a partir desta atividade.

Entrada: Informações (máquinas, diretórios, *urls*, usuário e senha para acesso, padrões de nomenclatura e domínio de valores dos campos das interfaces) sobre as interfaces externas e internas existentes no sistema legado e seu contorno.

Saída: Lista dos documentos existentes do sistema legado (especificações de requisitos, especificações de interfaces, visões arquiteturais, projeto de desenvolvimento).

6.2.2.2 Procedimento 2: Execução dos testes Caixa Preta (análise funcional das interfaces)

Descrição: Nesta atividade devem ser executados os testes Caixa Preta planejados na atividade anterior para cada interface identificada.

Entrada: Ficha de planejamento dos testes Caixa Preta.

Saída: Os resultados dos testes devem produzir novos casos de uso, que devem ser documentados.

Nas situações em que o caso de uso gerado pelo teste Caixa Preta já estiver documentado, ele deve ser validado e refinado mediante os resultados obtidos no teste.

6.2.3 Atividade 2.3 – Classificar e agrupar os requisitos levantados

As informações levantadas podem ser agrupadas de acordo com o domínio de negócio, a fim de facilitar a aplicação das técnicas de engenharia de software durante o trabalho de recuperação dos requisitos de software do sistema legado.

A seguir apresenta-se o procedimento detalhado desta atividade.

6.2.3.1 Procedimento 1: Análise dos resultados dos testes

Descrição: Após a execução dos testes Caixa Preta, os analistas devem realizar uma leitura dos resultados apresentados com o objetivo de levantar e agrupar os requisitos de acordo com o tipo de interface que os tratam.

Entrada: Resultados produzidos pela execução dos casos de testes planejados na atividade anterior.

Saída: Requisitos levantados e agrupados de acordo com o domínio de negócio que especificam.

6.2.4 Atividade 2.4 – Evolução das fichas de especificação dos requisitos funcionais

Em seguida, inicia-se a criação de fichas de requisitos seguindo os modelos apresentados nas atividades anteriores, como resultado da análise inicial dos documentos e das interfaces.

A ficha de especificação gerada no subprocesso 1 deve ser evoluída ao final do subprocesso 2 por meio do refinamento dos casos de uso já produzidos e da inclusão de novos casos de uso identificados no subprocesso 2.

A seguir, apresenta-se o procedimento detalhado desta atividade.

6.2.4.1 Procedimento 1: Especificação dos requisitos por meio de casos de uso

Descrição: Uma última leitura mais aprofundada dos resultados dos testes deve ser realizada com o objetivo de especificar cada Caso de Uso levantado. Nesta última atividade os analistas deverão recorrer à consulta dos desenvolvedores especialistas para esclarecimento de ambiguidades ou dúvidas oriundas da análise das documentações.

Entrada: Grupo de requisitos do domínio de negócio que será alvo do trabalho desenvolvido nesta pesquisa, diagrama de casos de uso construído nas atividades anteriores e listas dos casos de uso e dos atores.

Saída: A lista de Casos de Uso deve ser completada com o preenchimento do campo “Descrição” e os analistas devem criar os requisitos de software derivados da análise inicial, referenciando a nova fonte de conhecimento, ou seja, a ficha criada a partir do diagrama e das listas de casos de uso e atores produzidos durante este ciclo.

Ao final do processo é evoluído o artefato que documenta os requisitos de software levantados, por meio de diagramas de caso de uso e informações complementares, conforme resultados obtidos ao final do subprocesso 1.

6.3 Subprocesso 3- Recuperação de requisitos por meio da análise comportamental

Nesta seção são detalhadas as atividades que constituem o subprocesso3, apresentado na figura 8.

A Figura 11 representa as atividades envolvidas no subprocesso3 da pesquisa, considerando-se as entradas, saídas e mecanismos envolvidos em cada atividade.

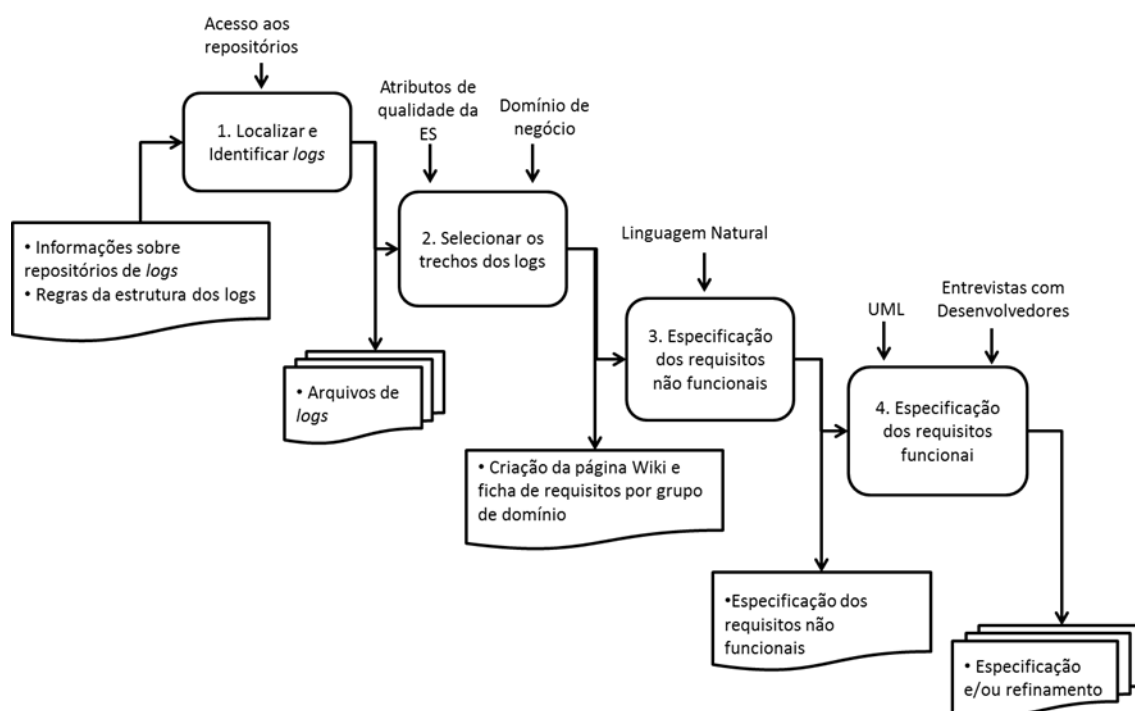


Figura 11: Atividades do processo de recuperação de requisitos por meio da análise comportamental (nos traços de execução – logs)

Fonte: Autores

6.3.1 Atividade 3.1 – Análise do comportamento do sistema para obtenção dos requisitos não funcionais

Após a conclusão dos testes Caixa Preta, nos quais foram produzidos estímulos ao sistema e seus resultados foram recolhidos, os traços de execução (logs)

gerados pelo sistema nestes testes devem ser analisados. O objetivo desta atividade é complementar a captura de requisitos, especialmente na perspectiva não funcional dos cenários construídos na *Atividade 2.2 – Análise das funcionalidades disponíveis nas interfaces* do subprocesso 2.

Dentre as análises realizadas nesta atividade, encontram-se as latências dos processos de negócio e a segurança na transmissão e armazenamento dos dados utilizados pelo sistema.

A seguir apresenta-se o procedimento detalhado desta atividade.

6.3.1.1 Procedimento 1: Localizar e identificar os logs

Descrição: Para se iniciar o subprocesso 3 a primeira atividade é identificar o repositório (ou os repositórios) onde os registros dos traços de execução (*logs*) produzidos pelo sistema legado são armazenados e localizá-los. Nesta atividade, é necessário também conhecer a estrutura da informação utilizada pelos desenvolvedores para a construção dos *logs* gerados. Esta informação deve ser obtida pelos manuais de boas práticas de desenvolvimento, se houver ou ainda pela consulta aos desenvolvedores.

Entrada: Informações das máquinas, diretórios, usuário e senha para acesso, padrões de nomenclatura dos documentos, padrões de estrutura dos documentos sobre os registros dos traços de execução do sistema legado.

Saída: Lista dos documentos/arquivos existentes do sistema legado com o registro dos *logs* e a estrutura da informação na qual se baseiam estes registros.

6.3.2 Atividade 3.2 – Classificando e agrupando a informação

A classificação e o agrupamento seguem os padrões utilizados na engenharia de requisitos, neste caso, os requisitos não funcionais, os quais ainda podem ser agrupados de acordo com o atributo de qualidade que se refere.

A seguir apresenta-se o procedimento detalhado desta atividade.

6.3.2.1 Procedimento 1: Selecionar os trechos dos logs

Descrição: Após a coleta dos registros referentes aos traços de execução (*logs*) produzidos pelo sistema legado, os analistas devem realizar uma leitura dos resultados apresentados com o objetivo de selecionar os trechos dos logs para levantar e agrupar os requisitos de acordo com os atributos de qualidade que tratam.

Entrada: Conteúdo os arquivos de logs localizados na atividade anterior.

Saída: Requisitos levantados e agrupados de acordo com os atributos de qualidade.

6.3.3 Atividade 3.3 – Evolução/Criação de fichas de especificação dos requisitos

Em seguida, inicia-se a evolução das fichas de requisitos funcionais, seguindo os modelos apresentados nas atividades anteriores, além da criação das fichas de requisitos não funcionais como resultado da análise comportamental por meio dos traços de execução do sistema legado.

A seguir, apresentam-se os procedimentos detalhados desta atividade (ver Figura 14, 3º e 4º blocos).

6.3.3.1 Procedimento 1: Especificação dos requisitos não funcionais (por meio de linguagem natural)

Descrição: Uma segunda leitura mais aprofundada nos traços de execução deve ser realizada com o objetivo de especificar os requisitos não funcionais registrados. Nesta atividade os analistas deverão recorrer à consulta dos desenvolvedores especialistas para esclarecimento de ambiguidades ou dúvidas oriundas da análise dos registros.

Os requisitos não funcionais serão documentados por meio de linguagem natural sem estruturação, no entanto, atendendo aos parâmetros e métricas padrões para cada tipo de atributo de qualidade analisado, quando for possível obter tais informações da fonte analisada.

Entrada: Registros dos traços de execução (*logs*) produzidos pelo sistema legado que será alvo do trabalho desenvolvido nesta pesquisa, devidamente agrupados e classificados, conforme atividade anterior.

Saída: Os analistas devem criar os requisitos não funcionais derivados da análise inicial, referenciando a nova fonte de conhecimento e a classificação do requisito, o resultado da atividade anterior.

Após a documentação dos requisitos não funcionais, a ficha de especificação dos requisitos funcionais, gerada e refinada pelos subprocessos 1 e 2, deve ser evoluída ao final do subprocesso 3 por meio do refinamento dos casos de uso já produzidos e da inclusão de novos casos de uso identificados no subprocesso 3.

6.3.3.2 Procedimento 2: Especificação dos requisitos funcionais (Evolução/Refinamento dos ciclos 1 e 2 por meio de casos de uso)

Descrição: Uma última leitura mais aprofundada dos resultados dos traços de execução (*logs*) deve ser realizada com o objetivo de refinar os Casos de Uso levantados nos processos anteriores. Nesta última atividade os analistas deverão recorrer à consulta dos desenvolvedores especialistas para esclarecimento de ambiguidades ou dúvidas oriundas da análise dos registros.

Entrada: Trechos de registros dos traços de execução que serão alvos do trabalho desenvolvido nesta atividade, conforme o agrupamento e classificação realizada na atividade anterior.

Saída: A lista de Casos de Uso deve ser completada com o preenchimento do campo “Descrição” e os analistas devem criar os requisitos de software derivados da análise inicial, referenciando a nova fonte de conhecimento.

7 Conclusões

A figura 12 resume a estrutura do processo, a partir dos subprocessos, atividades e procedimentos que o constituem.

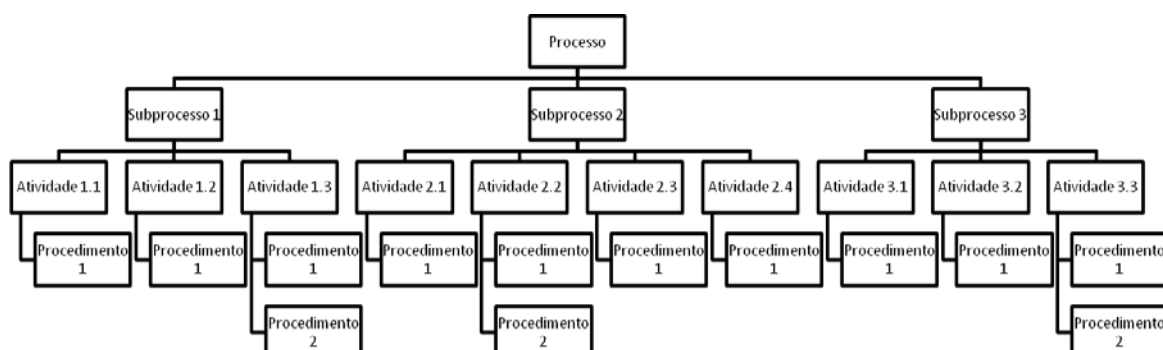


Figura 12: Estrutura do processo a partir dos subprocessos, atividades e procedimentos
Fonte: Autores

A tabela 1 resume as atividades e procedimentos executados em cada passo dos três ciclos da Pesquisa-Ação (PA), detalhados nas seções anteriores deste artigo.

Tabela 1: Resumo das atividades e procedimentos dos ciclos da PA
Fonte: Autora

Ciclos da PA	Passos da PA	Atividade	Procedimento
Ciclo 1 – Recuperação de requisitos por meio da análise de documentações	Coleta de Documentos	Atividade 1.1 – Análise das documentações disponíveis para obtenção dos requisitos funcionais e não funcionais	Procedimento 1: Identificação e localização dos documentos existentes
	Análise e Classificação	Atividade 1.2 – Classificando e agrupando os documentos e requisitos levantados	Procedimento 1: Agrupamento dos documentos e requisitos por domínio de negócio
	Registro dos Requisitos (UML)	Atividade 1.3 – Redocumentação dos requisitos levantados na atividade 1.1	Procedimento 1: Construção do Diagrama de Casos de Uso
			Procedimento 2: Especificação dos requisitos por meio de casos de uso
	Avaliação	REUNIÃO COM EQUIPE DE ANALISTAS DO SISTEMA LEGADO E CONCLUSÃO PARCIAL	

Ciclo 2 – Recuperação de requisitos por meio da análise de interfaces	Identificação das interfaces	Atividade 2.1 – Identificando o contexto de utilização do sistema e suas interfaces (sistemas externos e usuários)	Procedimento 1: Localizar e identificar os documentos existentes
	Análise e identificação das funções expostas	Atividade 2.2 – Análise das funcionalidades disponíveis nas interfaces	Procedimento 1: Desenhar e Planejar os testes Caixa Preta (análise funcional das interfaces)
	Estímulo das interfaces (Testes Caixa Preta)	Atividade 2.2 – Análise das funcionalidades disponíveis nas interfaces	Procedimento 2: Execução dos testes Caixa Preta (análise funcional das interfaces)
		Atividade 2.3 – Classificar e agrupar os requisitos levantados	Procedimento 1: Análise dos resultados dos testes
		Atividade 2.4 – Evolução das fichas de especificação dos requisitos funcionais	Procedimento 1: Especificação dos requisitos por meio de casos de uso
	Avaliação	REUNIÃO COM EQUIPE DE ANALISTAS DO SISTEMA LEGADO E CONCLUSÃO PARCIAL	
Ciclo 3 – Recuperação de requisitos por meio da análise comportamental	Identificação dos traços de execução (logs)	Atividade 3.1 – Análise do comportamento do sistema para obtenção dos requisitos não funcionais	Procedimento 1: Localizar e identificar os arquivos de registro dos logs
	Análise e classificação	Atividade 3.2 – Classificando e agrupando a informação	Procedimento 1: Agrupamento dos requisitos levantados por domínio de negócio
	Registro dos requisitos (Linguagem Natural e UML)	Atividade 3.3 – Evolução/Criação de fichas de especificação dos requisitos	Procedimento 1: Especificação dos requisitos não funcionais por meio de linguagem natural
			Procedimento 2: Evolução/Refinamento dos requisitos funcionais (ciclos 1 e 2) por meio de casos de uso
	Avaliação	REUNIÃO COM EQUIPE DE ANALISTAS DO SISTEMA LEGADO E CONCLUSÃO PARCIAL	

Visando a orientar as conclusões deste artigo, procura-se responder à seguinte questão relativa a esta pesquisa-ação:

- Como aplicar técnicas de Engenharia de Software para a obtenção dos requisitos de um sistema legado por meio de um processo para a recuperação destes requisitos?

Esta questão pode ser respondida utilizando-se o processo obtido na Figura 7, onde inicialmente aplicou-se o subprocesso 1 descrito pelas suas respectivas atividades indicadas na seção 6.1.

Verificou-se que somente a análise da documentação não era suficiente para a determinação dos requisitos do sistema legado, de modo que foi necessário realizar o segundo ciclo da Pesquisa-Ação, correspondente ao subprocesso 2.

Neste ciclo foram avaliadas principalmente as interfaces com os demais sistemas, por meio dos testes de caixa preta (análise funcional das interfaces), aplicando-se deste modo o subprocesso 2, com suas respectivas atividades descritas na seção 6.2.

Após a conclusão dos testes Caixa Preta do subprocesso 2, nos quais foram produzidos estímulos ao sistema e seus resultados foram recolhidos, os traços de execução (logs) gerados pelo sistema nestes testes foram analisados com o objetivo de complementar a captura de requisitos, especialmente na perspectiva não funcional dos cenários construídos nas sessões 6.2.2: Análise das funcionalidades disponíveis nas interfaces.

Assim, foi necessário o terceiro ciclo da Pesquisa-Ação, obtendo-se o subprocesso3 e suas respectivas atividades descritas na seção 6.3.

8 Futuros trabalhos

Propõe-se a aplicação do processo resultante a um sistema legado, cuja documentação seja praticamente inexistente, para se identificar a necessidade de técnicas adicionais relevantes às propostas neste trabalho, já que o subprocesso 1 terá pouca aplicabilidade neste contexto.

Além disso, sugere-se que sejam desenvolvidos trabalhos futuros que validem a qualidade dos artefatos produzidos na aplicação do processo proposto neste trabalho por meio da sua utilização no processo de desenvolvimento de software do novo sistema que deverá substituir o sistema legado analisado, com a engenharia avante (*forward*).

9 Referências bibliográficas

COUGHLAN, P.; COUGHLAN, D. Action research – action research for operations management. **International Journal of Operations & Production Management**, V.22 (No.2), 2002. 220-240.

CRUZ, T. **Workflow II: A tecnologia que revolucionou processo**. Rio de Janeiro: E-paper, 2004. 212 p.

DUCASSE, S.; POLLET, D. Software Architecture Reconstruction: A Process-Oriented Taxonomy. **IEEE TRANSACTIONS ON SOFTWARE ENGINEERING**, VOL. 35, NO. 4, p. 573-591, 2009.

, M.; NOWAKOWSKI, W. **Case-based Reuse with Partial Requirements Specifications**. 18th IEEE International Requirements Engineering Conference (RE). Vienna Univ. of Technol., Vienna, Austria: IEEE. 2010. p. 399-400.

KALSING, A. C. et al. **An Incremental Process Mining Approach to Extract Knowledge from Legacy Systems**. 14th IEEE International Enterprise Distributed Object Computing Conference. Vitória, ES, Brazil: IEEE. 2010. p. 79 -

88.

LIU, K.; ALDERSON, A.; QURESHI, Z. **Requirements Recovery from Legacy Systems by Analysing and Modelling Behaviour**. Washington DC, USA: ICMS, IEEE Computer Society. 1999.

MIGUEL, P. A. C. **Estruturação do desenvolvimento de produtos e implantação de um método de suporte: intervenção através da pesquisa-ação**. Universidade de São Paulo. São Paulo, p. 138. 2005.

MILLHAM, R. **Migration of a Legacy Procedural System to Service-Oriented Computing Using Feature Analysis**. International Conference on Complex, Intelligent and Software Intensive Systems. University of Bahamas, Bahamas and Durban University of Technology, South Africa: IEEE. 2010. p. 538 - 543.

POHL, K. **Requirements Engineering: Fundamentals, Principles, and Techniques**. [S.l.]: Hardcover, 2010.

PRESSMAN, P. D. R. S. **Software Engineering - A PRACTITIONER'S APPROACH**. New York: Mc Graw Hill, 2010.

RUB, F. A. A.; ISSA, A. A. **A Business Process Modelling Driven Approach for Legacy Systems Evolution**. [S.l.]: IEEE. 2009. p. 43-49.

ŚMIAŁEK, M. **Software Development with Reusable Requirements-Based Cases**. Warsaw. 2007.

SOMMERVILLE, I. **Software Engineering**. 8th Edition. ed. [S.l.]: [s.n.], 2010.

SOMMERVILLE, I.; KOTONYA, G. **Requirements Engineering: Processes and Techniques**. [S.l.]: Wiley, 1998.

THIOLLENT, M. **Metodologia da Pesquisa-Ação**. [S.l.]: Cortez, 2008.