HEITOR LUIZ ITAMI

Modelo para migração de sistemas legados

Monografia apresentada como trabalho final ao Programa de Educação Continuada em Engenharia (PECE) da Escola Politécnica da Universidade de São Paulo

São Paulo
2010

HEITOR LUIZ ITAMI

Modelo para migração de sistemas legados

Monografia apresentada como trabalho final ao Programa de Educação Continuada em Engenharia (PECE) da Escola Politécnica da Universidade de São Paulo

Área de Concentração:
Tecnologia da Informação

Orientador: Prof. Eduardo de Oliveira

São Paulo
2010

AGRADECIMENTOS

Ao professor Eduardo de Oliveira, pela orientação e pelo constante estímulo transmitido durante todo o trabalho.

À minha família pelo apoio e ajuda durante todo o curso, à minha namorada pela enorme compreensão e valoroso auxílio nos momentos difíceis e a todos que colaboraram direta e indiretamente, na execução deste trabalho.

Perder tempo em aprender coisas que não interessam, priva-nos de descobrir coisas interessantes.

(Carlos Drummond de Andrade)

RESUMO

Sistemas legados são sistemas baseados em computadores, que perduram por vários anos ou até mesmo décadas nas empresas, por executarem importantes funções, tornando-se vitais para o negócio. No entanto, as empresas podem ter a necessidade de substituí-los por um novo sistema. O presente trabalho trata de um modelo a ser utilizado nos processos de migração de sistema legado, que tem como objetivo guiar as empresas na verificação da viabilidade da migração e no levantamento das informações necessárias para a construção do novo sistema. Para a estruturação desse modelo, foi feita pesquisas bibliográficas sobre conceitos de sistemas legados, projetos e de engenharia de software. A necessidade do emprego de um modelo justifica-se devido aos altos custos que um projeto de migração pode alcançar e ao fato de que o novo sistema deve, não só trazer benefícios, como também manter os processos existentes funcionando corretamente. Através desse modelo pode ser verificado que, apesar do principal objetivo ser a substituição do sistema legado, este possui um importante papel durante o processo de migração.

Palavras-chave: Sistema legado. Migração. Modelo. Novo sistema.

ABSTRACT

Legacy systems are computer based systems, which last for several years or even decades in companies. They perform important functions and are vital to the business. However, companies may need to replace them for a new system. This study works with a model to be used in the migrating legacy system process, which aims to guide companies to verify the feasibility of migration and mapping the necessary information for construction of the new system. Literature searches on concepts of legacy systems, projects and software engineering was made to structure this model. The necessity of using a model is justified because the migration project can achieve a high costs and the new system must bring benefits, and also keep existing processes working properly. Working with this model, it is possible to infer that the legacy system has an important hole during the migration process, although the main objective being its substitution.

Keywords: Legacy system. Migration. Model. New system.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

As necessidades das empresas estão em constantes mudanças, seja por alterações na lei ou no próprio mercado que atuam. Para atender a essas mudanças, é comum a necessidade de modificações nos softwares que fornecem suporte ao negócio da empresa. Esses softwares podem ter sido desenvolvidos há décadas e alterados diversas vezes por diferentes equipes. Comumente tal fato resulta em sistemas sem especificação e sem documentação das regras de negócio, com um código sem padrão.

Esses softwares antigos, frequentemente referidos como software legado são, em muitos casos, os responsáveis pelo suporte de importantes funções nas empresas e acabam se tornando indispensáveis.

Há casos em que a empresa deseja migrar seu sistema legado por um novo. No entanto, como os legados normalmente desempenham importantes funções para a empresa, é arriscado substituí-los. Esses riscos podem ser mitigados com a utilização de um modelo para a migração do sistema legado, que será o tema do presente trabalho.

Nesse trabalho é descrito uma forma de auxiliar os gestores a decidirem se um projeto de migração deve ser iniciado ou não e como fazer o levantamento das informações necessárias para a construção do novo sistema que irá substituir o legado.

1.1 OBJETIVO

O presente trabalho tem como objetivo estabelecer um modelo para a migração de sistemas legados, com o intuito de prover o menor impacto possível para a empresa. Esse modelo será baseado nos métodos da engenharia de software, com foco no estudo de viabilidade, levantamento dos requisitos e as abordagens para se extrair as regras do sistema.

Como estudo de caso, será analisada uma seguradora que utiliza um sistema legado desenvolvido internamente e que há quase dez anos é responsável por todo o suporte do fluxo operacional da empresa. Essa empresa almeja utilizar um sistema mais moderno, feito com uma linguagem de programação atual e que seja mais flexível. O modelo proposto no trabalho será empregado nesse cenário, com o objetivo de se obter uma maior clareza dos benefícios da migração e de facilitar a construção do novo sistema.

1.2 JUSTIFICATIVA

Quando uma empresa opta por migrar um dos seus legados, não está alterando somente a tecnologia, mas está substituindo um sistema que conquistou a confiança da empresa e que possui problemas conhecidos, por um novo que pode gerar incertezas quanto ao seu comportamento.

Por isso, ao iniciar um projeto de migração, deve haver convicção de que a migração é realmente necessária e clareza do que o novo sistema deve oferecer. Caso contrário, a empresa corre o risco de despender grande quantidade de recursos e ao final ter um sistema inferior ao anterior.

As empresas que preferirem evitar esse risco podem postergar as alterações mais significativas em seus legados. No entanto, com o passar do tempo, os sistemas legados normalmente acabam precisando sofrer alterações significativas. Um dos motivos pode ser, por exemplo, alteração de leis que impactam o negócio.

Portanto, nos casos onde se opta por migrar um sistema legado, nota-se a importância de montar cuidadosamente a migração, em especial no início do projeto. A migração de um sistema legado sem um correto estudo de viabilidade e um completo levantamento dos requisitos pode resultar em uma grande perda financeira, o que pode até mesmo comprometer a continuidade da empresa, caso o novo sistema não atenda as necessidades do negócio e ainda impossibilite o retorno ao legado.

2 REVISÃO DA LITERATURA

A teoria abordada nesse capítulo é uma parte de toda a teoria disponível sobre o assunto, pois será descrito somente o que for necessário para fundamentar o modelo proposto.

2.1 SISTEMAS LEGADOS

Segundo afirma Sommerville (2007), sistemas complexos e de grande porte baseados em computadores exigem muito esforço e tempo para serem desenvolvidos e por isso normalmente possuem uma vida longa. Em casos de sistemas críticos de negócio, além de oneroso, é muito arriscado substituí-los e por isso são mantidos por longo período. Assim, estes sistemas continuam sendo desenvolvidos ao longo de suas vidas, para acomodar novos requisitos, solicitação de melhorias e etc.

Sommerville (2007) afirma que sistemas legados são sistemas baseados em computadores, que foram desenvolvidos no passado. Normalmente possuem tecnologia mais antiga ou obsoleta. Segundo Penteado apud Gandin (2003, p.13), os sistemas legados “[...] são aqueles sistemas que estão em uso por muito tempo, que atendem aos requisitos dos usuários e são de difícil substituição ou porque a reimplementação de seu código é inviável financeiramente ou porque eles são imprescindíveis.” Em outras palavras, conforme afirma Pressman (2006), as características dos sistemas legados são a longevidade e a grande importância para o negócio, pois normalmente são os responsáveis pelo suporte de funções essenciais ao negócio e tornam-se indispensáveis para a empresa. O autor também complementa com outra característica comum nos sistemas legados, que é a falta de documentação, exceto o código fonte.

Conforme afirma Sommerville (2007), apesar dos sistemas legados serem sistemas baseados em computadores, incluem processos e procedimentos antigos, além das partes de hardware e software. Esses processos e procedimentos são as

velhas formas de se executar as tarefas e que dificilmente são alteradas porque são baseadas em software legado. Pode-se afirmar que, alterar uma parte do sistema normalmente implica em alterações em outras partes. O autor ilustra os componentes de um sistema legado e seus relacionamentos, conforme pode ser visto na figura 1, e descreve o detalhamento separadamente.

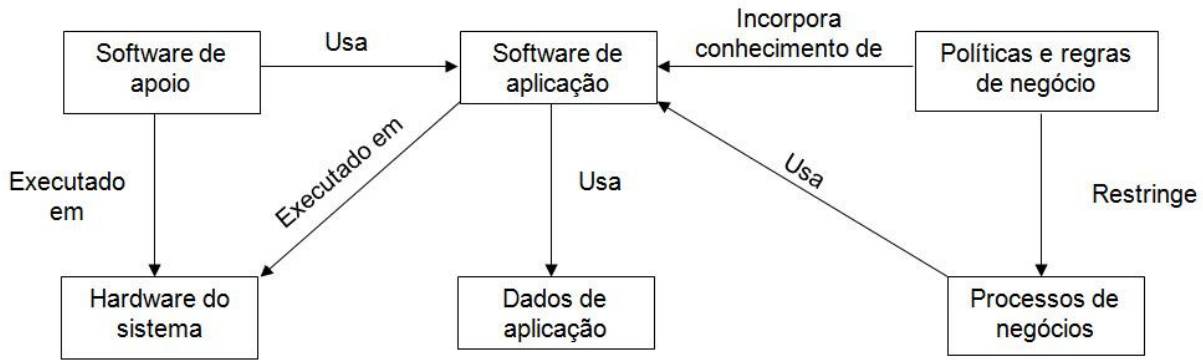

Figura 1 - Componentes de sistema legado (SOMMERVILLE, 2007).

O hardware do sistema legado muitas vezes é constituído de equipamentos ultrapassados. Há sistemas legados que foram escritos para serem executados em determinado hardware que pode não estar mais disponível no mercado e seja oneroso mantê-lo.

Software de apoio, como o próprio nome diz, são softwares que fornecem suporte ao sistema legado, como por exemplo, sistema operacional, utilitários fornecidos pelo fabricante do hardware e compiladores usados no desenvolvimento de sistemas. Esses softwares podem ser ultrapassados e não contarem mais com o suporte de seus fabricantes.

O software de aplicação é o responsável por fornecer os serviços de negócio e normalmente é formado por um conjunto de programas, que foram desenvolvidos em diferentes períodos. O termo sistema legado também é utilizado para identificar o software de aplicação ao invés de todo o sistema.

Os dados de aplicação são os dados processados pelo software de aplicação. É comum haver um grande volume de dados acumulados ao longo da vida do sistema legado. Esses dados podem apresentar inconsistências e estarem

duplicados em vários arquivos, diminuindo a confiabilidade das informações armazenadas.

Os processos de negócio são os procedimentos utilizados para se atingir um determinado objetivo de negócio, ou seja, o modo como são feitas as tarefas na empresa. Esses processos podem ser projetados com base no sistema legado existentes e serem restringidos pela funcionalidade fornecida. Nesses casos, se houver a necessidade de se alterar um processo de negócio, pode resultar também na modificação de outras partes do sistema.

As políticas e regras de negócios são as definições de como o negócio da empresa deve ser conduzido e as suas restrições. O uso do sistema de aplicação pode fazer parte dessas políticas e regras de negócios.

Segundo Gandin (2003, p. 14) “os sistemas legados constituem-se em causadores de sérios problemas às organizações, pois, sendo essenciais às atividades das mesmas, devem estar em manutenção permanente, implicando altíssimos custos”. Segundo Pressman (2006), a empresa pode optar por não fazer nenhuma modificação significativa no sistema legado, contanto que esteja atendendo as necessidades de seus usuários, funcionando confiavelmente e não precise ser consertado. No entanto, o autor afirma que com o passar do tempo, os sistemas legados frequentemente acabam evoluindo por pelo menos uma das razões abaixo:

> 1. O software precisa ser adaptado para satisfazer às necessidades do novo ambiente ou tecnologia computacional.
> 2. O software precisa ser aperfeiçoado para implementar (sic) novos requisitos do negócio.
> 3. O software precisa ser estendido para torná-lo interoperável com os sistemas ou bancos de dados mais modernos.
> 4. O software precisa ser rearquitetado para torná-lo viável em um ambiente de rede. (PRESSMAN, 2006, p. 9)

2.2 PROJETO

A migração de um sistema legado deve ser tratada como um projeto. Segundo o Project Management Institute (2004), responsável pelo Guia PMBOK, as características gerais de projeto são definidas como sendo: “Um projeto é um

esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo.” (PROJECT MANAGEMENT INSTITUTE, 2004, p. 5).

Segundo o Project Management Institute (2004), o termo temporário não significa que os projetos têm curta duração, mas sim que a duração de qualquer projeto é finita. Não existem projetos que os esforços sejam contínuos. O final do projeto é alcançado quando os objetivos deste são atingidos, quando ficar claro que os objetivos não serão ou não poderão ser atingido ou quando se concluir que o projeto não é mais necessário e for encerrado.

Conforme Project Management Institute (2004) descreve, o resultado de um projeto pode ser um produto quantificável, como por exemplo, um objeto produzido que pode ser o item final ou mesmo uma parte de um item maior. Outro resultado possível é a aquisição da capacidade de realizar um novo serviço. Um projeto também pode ter como entrega final o resultado de uma pesquisa ou a conclusão de um documento.

O Project Management Institute (2004) afirma que uma característica importante das entregas do projeto é a singularidade, ou seja, todo o projeto cria entregas exclusivas. Mesmo que dois projetos possuam elementos comuns, haverá características particulares em cada um que resultará na singularidade fundamental do trabalho dos projetos. Vargas (2007) complementa essa característica, quando comenta que um projeto é um empreendimento não repetitivo, ou seja, é um evento que não pertence à rotina da empresa.

Lins (2007) afirma que usualmente os projetos são divididos em fases, para melhorar o controle gerencial. O conjunto dessas fases é conhecido como ciclo de vida de um projeto, que engloba do início até o seu fim.

Lins (2007) assinala que cada fase é marcada pela entrega de um ou mais produtos. O produto de cada fase é considerado um subproduto do projeto como um todo. Essas entregas devem ser tangíveis e verificáveis.

Ainda segundo Lins (2007), as fases e os seus subprodutos devem seguir uma sequência lógica, garantindo um produto do projeto adequado. Ao final de cada fase, deve haver uma revisão dos produtos gerados e a avaliação do desempenho do projeto. Essas análises, denominadas saídas de fase ou passagens de estágio, têm como objetivos detectar erros e riscos, além de avaliar se o projeto deve prosseguir ou não.

Vargas (2007) destaca as vantagens em dividir o projeto em fases. Facilita verificar o que foi ou não feito no projeto, analisar como está o progresso e possibilita indicar em qual ponto o projeto se encontra em determinado momento.

Vargas (2007) afirma que a principal consideração a ser feita em relação ao ciclo de vida do projeto é o nível de esforço, onde se entende por esforço a quantidade de pessoas envolvidas, dispêndio de trabalho e de dinheiro, as complicações e preocupações. Segundo o autor “o nível de esforço destinado ao projeto inicia-se em praticamente zero e vai crescendo até atingir um máximo e, logo após esse ponto, reduz-se bruscamente até atingir o valor zero, representante do término do projeto” (VARGAS, 2007, p. 10). Na figura 2 é ilustrada a variação do esforço em relação ao tempo. No entanto, Vargas (2007) destaca que a localização do máximo pode variar de um projeto para outro.

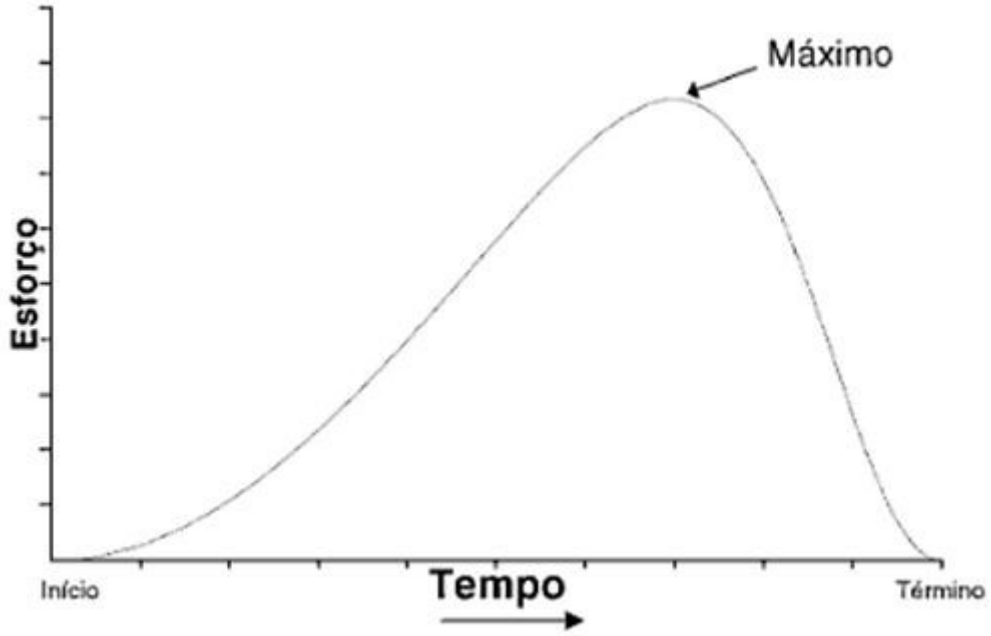

Figura 2 – Variação do esforço com o tempo para o projeto (VARGAS, 2007)

Segundo Vargas (2007, p. 12) “as fases do ciclo de vida do projeto dependem, intimamente, da natureza do projeto. Um projeto é desenvolvido a partir de uma idéia, progredindo para um plano, que por sua vez é executado e concluído”. No entanto, o autor afirma que genericamente, o ciclo de vida de qualquer projeto pode ser dividido em fases características, conforme se segue:

A primeira fase é a de iniciação, que é quando uma determinada necessidade é identificada e transformado em um problema estruturado que deve ser resolvido.

Nessa fase se define a missão e o objetivo do projeto, bem como é escolhida a melhor estratégia a ser seguida.

A segunda fase é de planejamento, que deve estar detalhado tudo aquilo que será realizado pelo projeto. O ideal é que o planejamento esteja em um nível de detalhe no qual não se terá dificuldades ou imprevistos para executá-lo.

A próxima fase é a de execução, onde se materializa tudo que foi planejado anteriormente. Nessa fase torna-se evidente qualquer erro que tenha sido cometido anteriormente. Grande parte do orçamento e do esforço é consumida nessa fase.

A fase de monitoramente e controle ocorrem paralelamente às fases de planejamento e execução. Tem como objetivo acompanhar e controlar o andamento do projeto, propondo ações preventivas e corretivas no menor tempo possível depois de detectada uma anomalia no projeto. O objetivo do controle é comparar o status atual do projeto com o que foi previsto no planejamento, tomando ações corretivas caso ocorra desvios.

A última fase é a de encerramento, onde a execução dos trabalhos é avaliada por uma auditoria interna ou externa, os livros e documentos do projeto são encerrados e todas as falhas ocorridas durante o projeto são analisadas e discutidas para que não ocorram novamente em novos projetos.

2.3 ESTUDO DE VIABILIDADE

Quando uma empresa inicia um projeto de migração de seu sistema legado, deve solicitar um estudo de viabilidade para que possa decidir se deve ou não prosseguir com o projeto. Caberá ao gerente de projeto decidir se o estudo de viabilidade será tratado como a primeira fase do projeto ou como um projeto autônomo separado.

Campos (2007) afirma que o objetivo do estudo de viabilidade é verificar se o projeto é possível e pode ser justificado. Em outras palavras, deve fornecer aos gestores a visão se o projeto é factível, se o produto final será benéfico, quais são as alternativas e se há uma alternativa preferível. O autor salienta que manter o sistema existente deve ser considerado como uma alternativa possível.

Segundo Sommerville (2007), a entrada de um estudo de viabilidade consiste em um conjunto preliminar de requisitos de negócios, um esboço da descrição do sistema e como o sistema pretende apoiar os processos de negócio.

Sommerville (2007) afirma que essa análise deve ser breve e focalizada, com o objetivo de responder a algumas questões. O novo sistema contribuirá para os objetivos gerais da organização? O novo sistema pode ser construído com a tecnologia atual, respeitando as restrições de custo e prazo definidas? O novo sistema pode ser integrado aos demais sistemas existentes na empresa?

A primeira questão sobre se o novo sistema irá contribuir ou não para os objetivos da empresa é crítica. Independentemente se o novo sistema irá substituir um sistema legado, este deve apoiar os objetivos da empresa, caso contrário não tem valor real. Tal afirmação pode parecer óbvia, mas há organizações que desenvolvem sistemas que não irão contribuir para seus objetivos. Conforme afirma Sommerville (2007, p. 97), isto pode ocorrer porque as empresas "[...] não têm um perfil claro desses objetivos, pois falham em definir os requisitos de negócios para o sistema, ou porque outros fatores políticos ou organizacionais influenciam na aquisição do sistema".

Sommerville (2007) afirma que para a realização do estudo de viabilidade é necessário identificar as informações necessárias para responder às três questões levantadas anteriormente. Posteriormente deve-se conversar com as fontes de informações, para que se descubram as respostas. As fontes de informações podem ser os gerentes dos departamentos em que o sistema será usado, engenheiros de software familiarizados com o tipo de sistema proposto, especialistas em tecnologia e usuários finais do sistema. Algumas questões que podem ser feitas a essas fontes de informações são:

- Quais seriam os impactos à empresa, caso o sistema legado fosse mantido e o novo sistema não fosse implantado?
- Quais são os problemas que se enfrenta com os atuais processos e como o novo sistema ajudaria a resolvê-los ou reduzi-los?
- Qual será a contribuição direta do novo sistema para os objetivos e requisitos da companhia?
- Os outros sistemas da organização podem receber e transferir informações para o novo sistema?

- O novo sistema exige tecnologia que ainda não foi empregada na empresa?
- O que deve e o que não deve ser compatível com o novo sistema?

Segundo Sommervile (2007), em posse das informações, é escrito o relatório de estudo de viabilidade, que deve constar a recomendação se o desenvolvimento do sistema deve ou não prosseguir. É possível também que no relatório constem propostas para mudança de escopo, orçamento, prazo e sugestões de novos requisitos de alto nível. Normalmente esse estudo necessita de duas a três semanas para ser concluído.

2.4 ANÁLISE DE REQUISITOS

A análise de requisito, segundo Braude (2004), consiste no processo de compreender e documentar o que uma aplicação destina-se a fazer depois de pronta. Em outras palavras, conforme descreve Kobryn et al (2005, p. 47) “a atividade de análise de requisitos procura descobrir o que os usuários e clientes de um produto de software querem que o sistema faça”.

Hora (2009) salienta a importância da análise de requisito, pois qualquer erro, inconsistência ou esquecimento pode resultar em mudanças em todo o projeto, atrasando a entrega e aumentando o seu custo.

Sommerville (2007) assinala que durante a análise de requisitos, é necessária a consulta a várias pessoas de uma organização. Segundo o autor, para se referir a qualquer pessoa ou grupo afetado pelo sistema, direta ou indiretamente é utilizado o termo *stakeholder*[1]. A definição segundo o autor é:

> Os *stakeholders* incluem os usuários finais que interagem com o sistema e todo o pessoal na organização que possa ser afetado por sua instalação. Outros *stakeholders* no sistema podem ser os engenheiros que estão desenvolvendo ou mantendo sistemas relacionados, gerentes de negócios, especialistas do domínio e representantes do sindicato. (Sommerville, 2007, p. 98)

De acordo com Sommerville (2007), diferentes *stakeholders* possuem requisitos distintos, que são expressos de diversas formas, o que dificulta o

[1] Há outra definição para o termo *stakeholder*. Segundo definição do dictionary.com, *stakeholder* é a pessoa ou grupo que investe ou é responsável por algo. No entanto, no presente trabalho será utilizada a definição dada por Sommerville.

entendimento dos engenheiros de requisitos. Além disso, o ambiente econômico e de negócios que se faz a análise é dinâmico e pode mudar durante o processo de análise, o que geraria a alteração ou novos requisitos. Novos requisitos também podem surgir de novos *stakeholders* que não haviam sido consultados anteriormente. Por isso, Previdelli (2001) afirma que na migração de sistema legado, é impossível conhecer todos os requisitos necessários do novo sistema e por isso deve ser projetado o mais flexível possível, possibilitando que novos requisitos sejam incorporados, sem a necessidade de grandes alterações em componentes já migrados.

Conforme explica Sommerville (2007), um modelo genérico análise de requisitos é composto por quatro atividades: obtenção de requisitos, classificação e organização de requisitos, priorização e negociação de requisitos e a documentação de requisitos. O autor salienta que esse modelo varia de empresa para empresa, possuindo uma versão ou modelo baseada nesse modelo genérico. O detalhamento de cada atividade, segundo Sommerville, é feito nos parágrafos seguintes.

A obtenção de requisitos é o processo de interação com os *stakeholders* para a obtenção dos requisitos do novo sistema. Em outras palavras, “a obtenção de requisitos é o processo que reúne informações sobre o sistema proposto e os existentes para obter os requisitos de usuários e de sistemas com base nessas informações” (SOMMERVILLE, 2007, p. 99).

A atividade de classificação e organização de requisitos consiste na atividade de se agrupar os requisitos relacionados e organizá-los em conjuntos coerentes.

Na priorização e negociação de requisitos, procura-se priorizar os requisitos, pois quando participam vários *stakeholders*, é inevitável que alguns requisitos sejam conflitantes. Por isso, essa atividade não só prioriza como também procura a resolução de conflitos de requisitos através de negociações.

Na documentação de requisitos, como o próprio nome diz, são gerados os documentos com os requisitos. Esses documentos podem ser formais ou informais.

Sommerville (2007) afirma que o processo de análise de requisitos é uma espiral, onde as atividades se intercalam à medida que o processo progride. Ou seja, esse processo deve ser um ciclo, ao finalizar a atividade de documentação de requisitos, inicia-se um novo ciclo com a obtenção de requisitos, tornando o processo iterativo. O entendimento dos analistas de requisitos aumenta a cada volta do ciclo.

Segundo Sommerville (2007), as fontes de informações consultadas durante a fase de obtenção de requisitos incluem documentação disponível, os *stakeholders* e especificações de sistemas similares ou de sistemas que interagem com o sistema que está sendo especificado.

No caso dos *stakeholders*, a interação ocorre por meio de entrevistas e observações. A observação consiste em inserir um analista de requisitos no ambiente de trabalho onde o novo sistema será utilizado, com o objetivo de observar o trabalho do dia a dia e anotar as tarefas dos funcionários dessa área.

Já as entrevistas com os *stakeholders*, conforme afirma Sommerville (2007), podem ser formais ou informais e fazem parte da maioria dos processos de engenharia de requisitos. Normalmente a equipe responsável pela obtenção dos requisitos formula questões sobre o sistema que o *stakeholder* está utilizando e sobre o sistema a ser desenvolvido. Os requisitos são derivados das respostas obtidas dessas questões. A importância dessa atividade é dada pelo autor quando afirma “as entrevistas são úteis para obter um entendimento geral sobre o que os *stakeholders* fazem, como eles podem interagir com o sistema e as dificuldades que enfrentam com os sistemas atuais” (SOMMERVILLE, 2007, p. 101).

Nas subseções seguintes serão detalhadas algumas técnicas desenvolvidas para dar apoio na atividade de obtenção de requisitos.

2.4.1 Pontos de vista

Segundo Sommerville (2007), os requisitos de sistema podem ser divididos em subconjuntos. Esses subconjuntos são formados pelos pontos de vista das várias fontes de requisitos. Conforme afirma o autor “cada ponto de vista fornece uma perspectiva nova do sistema, mas essas perspectivas não são completamente independentes – elas geralmente se sobrepõem de modo que haja requisitos comuns” (SOMMERVILLE, 2007, p. 100).

Sommerville (2007) afirma que os pontos de vista podem ser empregados para classificar os diferentes *stakeholders* e as demais fontes de requisitos. Há três tipos genéricos, os pontos de vista de interação, pontos de vista indiretos e os pontos de vista de domínio. Conforme salienta o autor, esses três pontos de vista

fornecem diferentes tipos de requisitos. Os tipos são descritos a seguir, segundo o autor.

Os pontos de vista de interação representam as pessoas ou sistemas que interagem diretamente com o sistema que está sendo especificado, como por exemplo, usuários finais e o sistema de banco de dados. Esse ponto de vista gera requisitos detalhados a respeito das características e as interfaces do sistema.

Os pontos de vista indiretos representam os *stakeholders* que não utilizarão diretamente o novo sistema, mas que exercem influência sobre os requisitos. Esses pontos de vista fornecem requisitos e restrições organizacionais de alto nível. Exemplos desses *stakeholders* pode ser o gerente da área que utilizará o sistema ou os responsáveis pela segurança da informação da empresa.

Os pontos de vista de domínio representam as características e restrições de domínio que possuem influência sobre os requisitos de sistema. Um exemplo que pode ser citado são os padrões existentes para que o novo sistema consiga se comunicar com os outros já existentes.

2.4.2 Casos de uso

A definição de caso de uso, segundo Pressman (2006), é ”[...] uma história estilizada sobre como um usuário final (alguém desempenhando um entre vários papéis possíveis) interage com o sistema sob um conjunto específico de circunstâncias” (Pressman, 2006, p. 129). O autor afirma que essa história pode ser um delineamento das tarefas ou interações, um texto narrativo ou uma representação diagramática. Sampaio (2007) complementa, salientando que um diagrama de caso de uso descreve cenários, que modelam uma funcionalidade de um sistema, do ponto de vista do usuário.

Sommerville (2007) afirma que os casos de uso constituem uma técnica para análise de requisitos e que se tornaram uma característica fundamental da notação Unified Method Language (UML). A definição de UML, segundo FERREIRA (2001), consiste em uma linguagem de modelagem padronizada voltada para engenharia de sistemas orientados a objetos. O autor complementa a definição da UML, com a seguinte afirmação:

> É a descrição básica e formal da linguagem de modelagem, situando-a como padrão para a análise e desenvolvimento de sistemas orientados a objeto [...] uma linguagem de modelagem bem definida deve possuir a capacidade de representar sistemas de qualquer tipo e grau de complexidade, sendo esta a função primordial da padronização oferecida pela UML. (FERREIRA, 2001)

Pressman (2006) afirma que o primeiro passo ao iniciar um caso de uso é definir os atores envolvidos na história. Segundo o autor, os atores são pessoas ou dispositivos que utilizam o sistema ou produto, dentro do contexto da função e comportamento que estão sendo descritos. Os atores se comunicam com o sistema e representam os papéis que as pessoas ou dispositivos desempenham quando o sistema está em operação. Os atores são externos ao sistema e possuem um ou mais objetivos ao utilizarem-no.

No caso de uso, o ator e um usuário final podem não ser a mesma coisa. Um usuário final pode desempenhar vários papéis quando utiliza um sistema, já um ator, que pode ser ou não uma pessoa, representa apenas um papel no contexto de um caso de uso.

Conforme afirma Scott (2002), no caso de uso, os atores podem ser representados como uma figura de linha, com um nome descritivo curto. Na figura 3 é exemplificada a representação de atores humanos e não humanos.

Figura 3 - Exemplos de representação de atores (Scott, 2002)

A representação de um caso de uso é feito através de uma elipse, possuindo um nome curto, que contém um verbo ativo e geralmente acompanhado por um substantivo. O nome do caso de uso pode aparecer tanto dentro como abaixo da elipse.

Scott (2002) afirma que a representação do relacionamento entre os atores e os casos de uso é feito através de diagrama de caso de uso. Segundo Booch ; Rumbaugh e Jacobson (2000), os atores e os casos de uso são conectados através da associação, o que indica a comunicação entre eles. A figura 4 ilustra um exemplo de diagrama de caso de uso básico com associação.

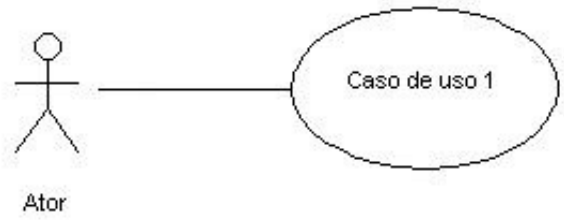

Figura 4 – Exemplo de diagrama de caso de uso com associação (Souza, 2006)

Conforme esclarece Scott (2002), com o objetivo de simplificar os casos de uso é possível estruturar os diagramas de casos de uso. Segundo o autor, a UML oferece três construções para decompor o comportamento comum e caminhos variantes.

A primeira é o relacionamento include ou inclui. Segundo Souza (2006), este relacionamento é utilizado quando um caso de uso precisa de ou é composto de outro. É representado por um arco pontilhado com o rótulo <<inclui>> ou <<include>>. A figura 5 ilustra esse relacionamento.

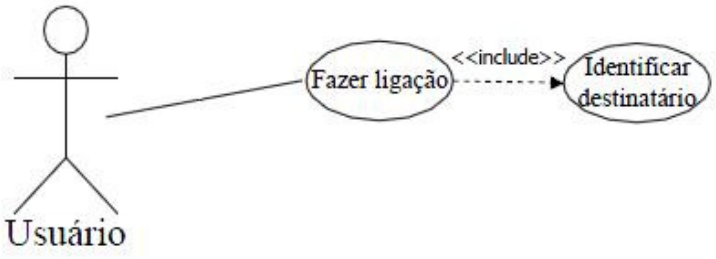

Figura 5 – Representação do relacionamento include (SOUZA, 2006)

O segundo relacionamento é o extend ou extensão. Souza (2006) esclarece que o relacionamento extend é utilizado quando um caso de uso pode opcionalmente utilizar outro. Scott complementa, quando afirma que “um meio de usar extends é criar um novo caso de uso em resposta a um curso alternativo de ação que tem vários passos associados” (SCOTT, 2002, p. 47). Na figura 6 é ilustrado esse tipo de relacionamento.

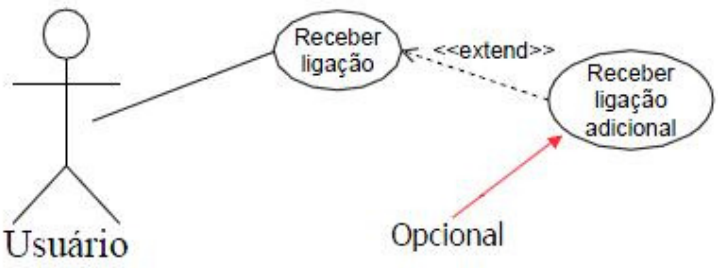

Figura 6 - Representação do relacionamento extend (SOUZA, 2006)

A terceira construção é a generalização. Segundo Santander e Castro (2004), a generalização pode ser empregada tanto entre casos de uso como entre atores. No caso dos atores, a generalização é empregada quando um ator pode ser visto como outro ator, mas que possui alguns comportamentos específicos. Para os casos de uso, conforme afirma Scott (2002, p. 47) “um caso de uso pai define um comportamento que seus filhos podem herdar, e os filhos podem adicionar ou sobrescrever este comportamento”. Na figura 7 é ilustrado um exemplo de generalização de casos de uso.

Figura 7 - Representação de generalização de casos de uso (SOUZA, 2006)

Segundo Scott “o caso de uso deve descrever um aspecto de utilização do sistema sem presumir qualquer projeto ou implementação (sic) específica. Em outras palavras, descreve o que o sistema necessita fazer sem especificar como o sistema executará isso” (SCOTT, 2002, p. 38, grifo do autor).

Conforme afirma Satzinger; Jackson e Burd (2009), o objetivo da descrição de um caso de uso é detalhar a interação entre o ator e o sistema. Chiarello e Mayer (2009) complementam, salientando que a descrição de um caso de uso deve ser isenta de detalhes ligados a interface com o usuário e aspectos ligados à arquitetura ou a codificação.

Ainda segundo Chiarello e Mayer (2009), os fluxos de eventos que devem constar na descrição do caso de uso são o fluxo principal e os alternativos ou exceções.

Conforme citado anteriormente, o ator ao utilizar o sistema possui um objetivo, no qual espera que o sistema o ajude a atingi-lo. O fluxo principal descreve o que ocorre quando o caso de uso é executado, detalhando os passos que levam o ator a alcançar seu objetivo.

Durante a definição dos passos do fluxo principal, podem ser identificados passos alternativos que devem ser detalhados no fluxo alternativo.

Segundo Chiarello e Mayer (2009), o fluxo de eventos alternativos pode ser entendido como desvios do fluxo principal, no qual alguns retornarão ao fluxo principal e outros finalizarão a execução do caso de uso. Conforme é detalhado o fluxo principal, deve-se analisar a existência de fluxos alternativos. Conforme Chiarello e Mayer (2009, p. 6) afirmam, essa análise pode ser feita com perguntas como “Existem respostas diferentes, dependendo da ação do ator? [...] O que poderia dar errado?”.

Conforme descreve Chiarello e Mayer (2009), na descrição do caso de uso devem constar o detalhamento da pré-condição e pós-condição. Uma pré-condição é a restrição existente sobre o início do caso de uso, ou seja, descreve o estado que o sistema deve estar para que o caso de uso possa ser iniciado. A pós-condição lista os possíveis estados que o sistema se encontrará ao término do caso de uso.

Satzinger; Jackson e Burd (2009) afirmam que um caso de uso completo também deve conter um identificador do caso de uso, uma breve descrição e a existência de requisitos especiais, ou seja, requisitos não funcionais relativos ao caso de uso descrito.

2.5 ENGENHARIA REVERSA

Segundo Chikofsky apud Vasconcelos (2007, p. 36) a definição de engenharia reversa é “o processo de análise de um sistema, a fim de identificar os componentes desse sistema e seus relacionamentos e criar representações do sistema em outra forma ou em um nível mais alto de abstração”. Vasconcelos (2007) afirma que o objetivo da engenharia reversa é a extração da documentação a partir do código fonte, visando facilitar a compreensão de um software existente.

Pressman (2006) relata que a documentação desatualizada ou inexistente dos sistemas legados e a mobilidade do pessoal de software são os motivos do desconhecimento do funcionamento do sistema legado pelas empresas. Segundo o autor, é provável que as pessoas responsáveis pela criação do sistema legado não estejam mais na empresa, já que esses sistemas normalmente possuem mais de 10

anos. Além disso, as gerações subseqüentes do pessoal de software que modificaram o sistema também podem não estar mais na organização.

Earls A.B.; Embury S. M.; Turner N. H (2002) esclarecem a importância da engenharia reversa na migração de um sistema legado. “Uma das principais barreiras para o sucesso da migração de um sistema legado é a falta de informação sobre o sistema a ser migrado. Com a complexidade dos modernos sistemas de computadores, são raros os casos em que a empresa tenha completo entendimento sobre seus sistemas. Isso é particularmente verdadeiro quando as regras de negócio estão contidas nesses sistemas, que frequentemente estão mal documentados. As regras de negócio normalmente não são completamente compreendidas pelos donos do sistema, por seus usuários e os responsáveis por sua manutenção. No entanto, essas regras de negócio devem ser migradas, juntamente com as outras funcionalidades do sistema legado” (EARLS A.B.; EMBURY S. M.; TURNER N. H., 2002, tradução nossa).

Segundo Vasconcelos (2007) a má documentação é resultado das alterações feitas no sistema legado ao longo dos anos de sua existência, sem que houvesse a atualização da documentação em paralelo. Por isso, o autor afirma “assim, o código fonte acaba se tornando a única fonte de informação atualizada (VASCONCELOS, 2007, p. 36). No entanto, o código se encontra em um nível baixo de abstração, pois são voltados para os compiladores, e os sistemas legados normalmente possuem milhares de linhas de códigos, o que dificulta o seu entendimento sem o apoio de ferramentas da engenharia reversa.

Existem diversos softwares no mercado que geram documentos a partir do código fonte. No entanto, não é objetivo do presente trabalho descrever sobre a utilização de uma dessas ferramentas. Ao invés disso, será exposta uma técnica para criação de modelos de objetos a partir da análise do código fonte.

Recchia e Penteado (2002) descrevem um dos métodos desenvolvidos para auxiliar o engenheiro de software na engenharia reversa e que é resumida nesse trabalho. Nesse método, o engenheiro de software tendo o primeiro contato com o sistema legado, deve ler parte do código fonte por aproximadamente uma hora sem interrupções, como telefonemas, conversas ou barulhos. Nesse período deve anotar o mínimo possível para que tenha o máximo de contato com o código. Após esse período, deve montar um relatório com as constatações feitas.

Se houver, também deve ser lida a documentação disponível. Apesar de o sistema legado ser alterado com o passar do tempo e provavelmente a maioria dessas alterações não estarem documentadas, o engenheiro de software pode conhecer como o sistema se comportou no passado e compreender o funcionamento do sistema.

No método descrito por Recchia e Penteado (2002), o engenheiro de software também deve obter a idéia inicial das funcionalidades do sistema observando-o em operação e entrevistando a pessoa que o está demonstrando. O engenheiro de software deve ter acesso às pessoas chaves da empresa, como os usuários, gerentes e os responsáveis pela manutenção do legado. Para obter um resultado satisfatório, devem ser entrevistados vários tipos de usuários. Nos casos de sistemas complexos, essas entrevistas podem ser intercaladas com a leitura do código e da documentação.

O próximo passo do engenheiro de software após o entendimento do sistema e refinar progressivamente um modelo de objetos de acordo com o código fonte. Desta forma, tendo uma idéia sobre o que o sistema representa, o engenheiro de software pode imaginar como modelar o sistema através de sua experiência. Deve ser idealizado um modelo hipotético de classes que represente o legado. Os nomes das classes, operações e atributos devem ser baseados nas convenções de nomenclatura do legado e na experiência do engenheiro de software.

Ainda segundo Recchia e Penteado (2002), o modelo de objetos obtido deve ser adequado ao sistema de banco de dados do sistema legado. Para isso, cada tabela do banco de dados relacional deve ser considerada como uma classe, preservando seu nome. Os atributos dessas classes são os nomes das colunas das tabelas.

Como último passo do método é o detalhamento das classes, em que se identificam os relacionamentos entre elas.

2.5.1 Modelos de objetos

Segundo Ricarte (2001, p. 6) “um modelo de objetos busca capturar a estrutura estática de um sistema mostrando os objetos existentes, seus

relacionamentos, atributos e operações que caraterizam cada classe de objetos". O autor também afirma que o uso deste modelo enfatiza o desenvolvimento em termos de objetos, permitindo uma representação mais próxima do mundo real.

Sommerville (2007, p. 121) define uma classe de objeto da seguinte maneira "[...] uma abstração de um conjunto de objetos que identifica atributos comuns [...] e os serviços ou operações fornecidas por cada objeto. Os objetos são entidades executáveis com os atributos e serviços de uma classe de objetos". Os objetos são instâncias das classes e podem ser criados vários objetos baseados na mesma classe. O autor afirma que os modelos desenvolvidos normalmente enfocam as classes de objetos e os relacionamentos entre elas.

Sommerville (2007) afirma que uma classe de objeto em UML é representada como um retângulo vertical com três seções. Na seção superior está o nome da classe de objetos. Na seção intermediária estão os atributos e na seção inferior do retângulo constam as operações associadas à classe de objeto. Na figura 8 é ilustrada uma classe, de nome Pessoa.

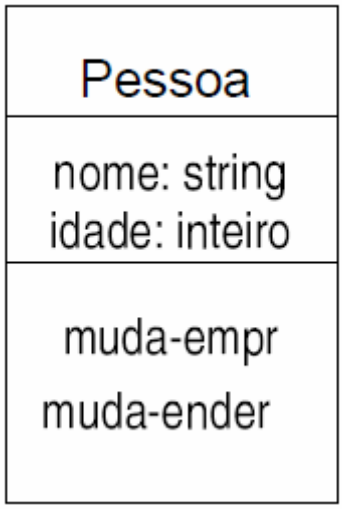

Figura 8 – Representação diagramática de uma classe com atributos e operações (RICARTE, 2001)

Conforme Sommerville (2007) afirma, as classes são organizadas em uma taxonomia, que segundo o autor "uma taxonomia é um esquema de classificação que mostra como uma classe de objeto está relacionada a outras classes por meio de atributos e serviços comuns" Sommerville (2007, p. 121).

Sommerville (2007) descreve que para apresentar a taxonomia, as classes são estruturadas em uma hierarquia de herança, onde as classes de objetos mais

genéricas, também conhecidas como classe-pai ou superclasse, ficam no topo da hierarquia. Os objetos mais especializados, também chamados de classe-filha ou subclasse, herdam seus atributos e serviços, podendo também possuir seus próprios atributos e serviços.

Segundo Sommerville (2007), na notação UML, a herança é mostrada no sentido ascendente, ou seja, a seta aponta da subclasse para a superclasse. Na figura 9 é exemplificada uma hierarquia de herança de um modelo dos usuários de uma biblioteca.

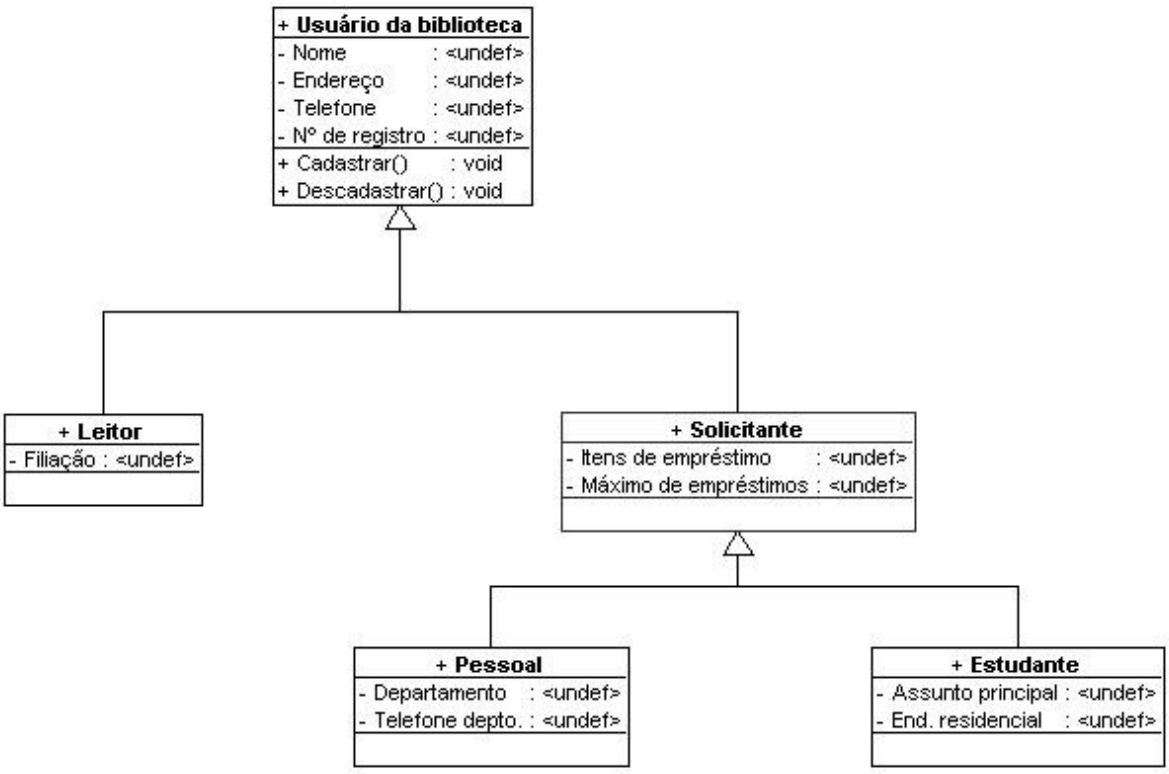

Figura 9 – Hierarquia de classe de usuário (SOMMERVILLE, 2007)

Outro relacionamento existente é a associação. Segundo Kon (2005, p. 17) a associação “é uma relação estrutural nas quais classes ou objetos estão interconectados”. Silva (2006) esclarece que a associação possui um nome, escrito junto à linha que representa o relacionamento, que normalmente é um verbo, mas também pode ser empregado um substantivo. O autor também afirma que “pode-se também colocar uma seta no final da associação indicando que esta só pode ser usada para o lado onde a seta aponta. Mas associações também podem possuir

dois nomes, significando um nome para cada sentido da associação" (SILVA, 2006, p. 7). Na figura 10 é exemplificada duas classes que se relacionam por associação.

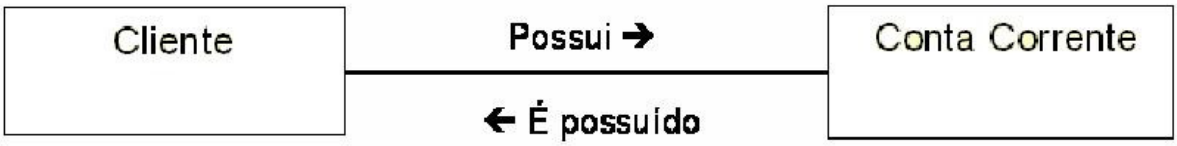

Figura 10 – Exemplo de um relacionamento entre classes Cliente e Conta Corrente que se relacionam por associação (SILVA, 2006, p. 8).

Segundo Silva (2006), para expressar a multiplicidade de um relacionamento, indica-se o número ou o intervalo de quantos objetos estão relacionados. O intervalo é representado escrevendo o primeiro número, seguidos de dois pontos consecutivos e o último número do intervalo. Assim, o intervalo de zero a um é escrito 0..1. Já o intervalo de um ou mais é representado por 1..* e o de zero ou mais como 0..* ou simplesmente *.

Sommerville (2007) esclarece que para modelar o comportamento de objetos, deve ser mostrado como as operações dos objetos são usadas. Em UML, os comportamentos podem ser modelados utilizando casos de uso, descritos na subseção 2.4.2 deste trabalho.

Em relação ao detalhamento necessário, Booch; Rumbaugh e Jacobson (2000, p. 103) afirmam que em cada diagrama deve ser revelado "[...] somente os detalhes suficientes para esclarecer os aspectos desejados. Informações irrelevantes podem desviar a atenção do leitor em relação ao ponto-chave que você deseja esclarecer". Em outras palavras, o detalhamento dos diagramas pode variar, dependendo do propósito e das necessidades do público alvo.

3 MODELO PROPOSTO

Nesse capítulo será proposto um modelo baseado na engenharia de software, que procura levantar as informações necessárias para a migração de um sistema legado. Esse modelo parte do pressuposto que os envolvidos no projeto terão acesso ao código fonte, ao banco de dados, documentação disponível, as equipes que utilizam e fazem a manutenção do sistema legado e aos gestores da empresa, pois serão as fontes de informações utilizadas.

O objetivo principal desse modelo é guiar na busca pela construção de um novo sistema, que seja superior ao que está sendo substituído, mas que não coloque em risco a confiabilidade e as operações da empresa.

Com o intuito de facilitar a compreensão do modelo proposto, cada passo constará nas subseções desse capítulo, com exceção das duas últimas subseções, que se referem à utilização do modelo como um projeto e a ilustração do quadro do modelo.

3.1 ESTRUTURAÇÃO DO PROBLEMA E DEFINIÇÃO DOS OBJETIVOS

Para uma empresa que perceba a necessidade de migrar seu sistema legado, o primeiro passo, do modelo proposto no presente trabalho, é estruturar os problemas do sistema atual, que levaram a empresa a almejar a sua substituição. Essas informações devem ser extraídas através de entrevistas com os gestores, usuários e os responsáveis pela manutenção do sistema legado.

Nessa análise devem constar, juntamente com os problemas, os impactos causados ao negócio, tanto na parte técnica como no operacional.

Com os problemas identificados, é possível definir os objetivos que a empresa espera atingir com a migração, para que esses problemas sejam resolvidos ou atenuados.

3.2 ESTUDO DE VIABILIDADE

O passo seguinte proposto no modelo é o estudo de viabilidade. Esse estudo é necessário devido à grande importância que normalmente os sistemas legados apresentam e os custos serem normalmente elevados para a construção de um sistema substituto.

As entradas para o estudo de viabilidade são os problemas e os objetivos levantados no passo anterior, juntamente com os requisitos de negócio da empresa.

Nesse estudo, as vantagens esperadas com a migração são esclarecidas e precisam justificar a necessidade de substituição do sistema legado. Conforme visto na parte teórica, no estudo de viabilidade deve constar como o novo sistema irá contribuir para que a empresa alcance seus objetivos, especificados nos requisitos de negócio.

No estudo também é importante que se contemple a alternativa de abortar a migração e manter o sistema atual. Nesse ponto, devem ser analisadas quais medidas serão necessárias para que o legado resolva ou mitigue os problemas levantados. Também devem ser detalhados os riscos que essa alternativa poderia acarretar.

O objetivo do estudo de viabilidade é auxiliar os gestores a decidirem se o projeto de migração é factível e benéfico para a empresa. Os gestores precisam avaliar se os benefícios esperados com a migração trarão vantagens competitivas frente aos concorrentes e se é esperado uma redução nos custos ou uma melhora nos processos, devido ao novo sistema. Esses pontos devem ser confrontados com o custo estimado para a migração.

Com essas informações, os gestores devem decidir se o custo benefício da migração é melhor do que quando comparado com a alternativa de se manter o sistema legado, onde deve ser considerado não só o esforço necessário para adaptá-lo as necessidades da empresa, como também os riscos existentes em se manter o legado.

3.3 MAPEAMENTO DO SISTEMA LEGADO

Se a decisão dos gestores for pela continuidade da migração, o próximo passo é o levantamento das informações necessárias para a construção do novo sistema. No modelo proposto, uma fonte de informação que deve ser explorada é o sistema legado.

Conforme visto na revisão bibliográfica, normalmente o sistema legado é responsável pelo suporte a funções essenciais ao negócio. Utilizá-lo como fonte de informações justifica-se devido ao fato de que o novo sistema deve continuar executando essas funções corretamente, garantindo a continuidade das operações da empresa.

Na figura 1, do capítulo 2.1, foi ilustrado os componentes que formam um sistema legado. Se a migração substituir somente parte dos componentes do legado, é necessário o mapeamento de todo o sistema.

Esse passo é importante, pois os componentes do sistema legado são interligados e a alteração em um deles provavelmente implicará na necessidade de se modificar os demais. As fontes de informações a serem utilizadas são os responsáveis pela manutenção do sistema legado e a documentação existente.

Portanto, no mapeamento deve ser listado o que existe em cada componente e analisado os impactos esperados em cada um com a migração. Para facilitar a visualização, a figura 1 deve ser adaptada para o sistema estudado.

3.4 ENGENHARIA REVERSA

Após o mapeamento dos componentes do sistema legado, é necessário compreender como o programa que será substituído trabalha. Na revisão bibliográfica foi afirmado que normalmente há regras de negócio que não são completamente compreendidas pelas empresas e que a única fonte de informação confiável e atualizada é o código fonte do sistema.

Portanto, para se extrair as regras de negócio existentes do sistema legado, é de vital importância o emprego da engenharia reversa. O objetivo desse passo é documentar essas regras em um nível de abstração mais alto do que o código fonte.

No modelo proposto, a documentação a ser gerada é constituída de casos de uso e modelo de objetos, conforme descrito na revisão bibliográfica.

O mapeamento do sistema legado e a engenharia reversa são atividades que se complementam na busca pela completa compreensão do sistema atual. Esse entendimento tem sua importância justificada no fato de que essas informações devem ser consideradas na construção do novo sistema, que deve substituir o legado, sem que haja uma queda na qualidade dos serviços e processos da empresa, seja por descumprimento a alguma regra ou a descontinuidade de alguma atividade devido à falta de integração entre os componentes do sistema.

3.5 ANÁLISE DE REQUISITOS

O próximo passo do modelo é a análise de requisitos, que tem como objetivo o levantamento e compreensão dos requisitos do novo sistema. As técnicas empregadas sãs as que foram descritas na revisão bibliográfica.

Esse passo deve ser feito após o mapeamento do sistema legado e a engenharia reversa, pois muitos dos requisitos para o novo sistema serão solicitações de melhorias de funcionalidades existentes no sistema legado. Portanto, para o engenheiro de software, conhecer a importância dessas funcionalidades, para que servem e como são utilizadas facilitaria as entrevistas com os stakeholders, além de auxiliar em um melhor entendimento dos requisitos.

Diferentemente do que é explicado na revisão bibliográfica, no modelo proposto, não só a atividade de análise de requisitos deve ser tratada como um ciclo iterativo, mas também os passos de mapeamento do sistema legado e a engenharia reversa devem fazer parte desse ciclo.

No primeiro ciclo deve ser feita as análises de mais alto nível. Ao término desse ciclo, os requisitos levantados para o novo sistema devem ser considerados para uma nova análise dos componentes do sistema legado e da engenharia reversa.

Os componentes do sistema legado que serão afetados pelos requisitos levantados, devem ser mais bem analisados, com o intuito de se detalhar as alterações necessárias. Já para as funcionalidades existentes no sistema legado e que houve solicitações de alterações ou melhorias, precisarão ser descritos em níveis mais baixos para que o novo sistema siga as regras vigentes.

O objetivo de se trabalhar na forma de ciclo iterativo é alcançar de forma progressiva uma maior compreensão de como o novo sistema deve trabalhar. O nível de detalhamento a ser alcançado deve ser de acordo com a necessidade dos engenheiros de software.

É importante salientar que existem ferramentas no mercado construídas para automatizar esse detalhamento. No entanto, conforme comentado anteriormente, nenhuma dessas ferramentas serão descritas no presente trabalho.

3.6 EMPREGO DO MODELO COMO PROJETO

O modelo proposto nesse trabalho pode ser tratado como um projeto, dentro do projeto de migração do sistema legado. Conforme descrito na revisão bibliográfica, o projeto é um esforço temporário feito para se criar um produto, serviço ou resultado exclusivo. No caso do projeto utilizando o modelo, que será chamado de projeto do modelo, o produto final esperado é a geração de toda a informação necessária para a construção do sistema que irá substituir o sistema legado.

No projeto do modelo, a fase de iniciação deve conter os dois primeiros passos do modelo, que são a estruturação do problema e definição dos objetivos da migração, juntamente com o estudo de viabilidade.

Na segunda fase do projeto do modelo, deve ser feito o planejamento de como os demais passos do modelo serão executados. Nesse planejamento deve ser definida a quantidade de pessoas alocadas para cada passo, o cronograma e o custo estimado. Para esses pontos a empresa deve considerar a sua disponibilidade de recursos e urgência para a migração.

Na fase de execução, os passos de mapeamento do sistema legado, engenharia reversa e análise de requisitos são trabalhadas de acordo com o cronograma planejado. Nessa fase se espera atingir o máximo do nível do esforço do projeto do modelo.

Na fase de encerramento, toda a documentação gerada nesse projeto precisa ser avaliada, com o objetivo de se certificar que as informações necessárias para a construção do novo sistema foram corretamente levantadas.

3.7 QUADRO DO MODELO

Com o objetivo de permitir a visualização do fluxo, dos objetivos e as saídas de cada passo, foi montado o quadro do modelo proposto, que consta na figura 12.

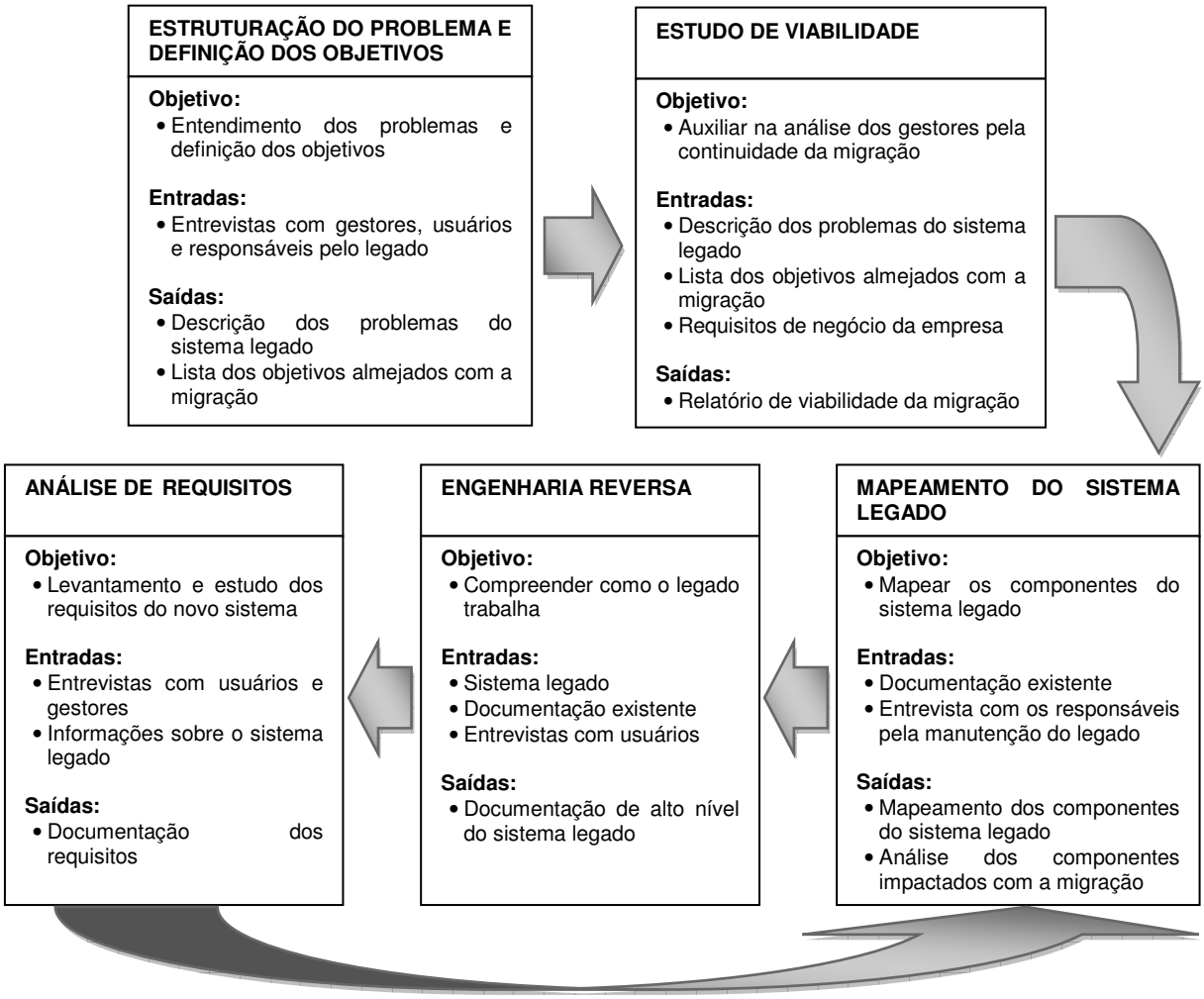

Figura 11 – Esquema do modelo de migração de sistemas legados

4 APLICAÇÃO DO MODELO EM UMA SEGURADORA

Nesse capítulo será primeiramente feito uma descrição da empresa e do sistema legado que serão analisados no presente trabalho. Posteriormente, serão detalhados alguns conceitos de seguros, com o objetivo de facilitar o entendimento do cenário. Após o esclarecimento desses pontos, os passos do modelo proposto serão aplicados na migração desse sistema legado.

4.1 O CENÁRIO

O modelo descrito no capítulo anterior será aplicado em uma seguradora, que almeja migrar seu sistema legado. Esse sistema é de grande importância para a empresa, pois é o responsável pelo controle das operações de todos os seguros da companhia.

Essa seguradora pertencente a um banco multinacional e está operando no Brasil há mais de dez anos. O seu principal produto é o de proteção financeira, no qual trabalha em parceria com grandes redes de lojas e hipermercados. Esse produto é vendido através dessas empresas varejistas, que ao venderem uma mercadoria parcelada, oferecem aos seus clientes um seguro, no qual caso ocorra desemprego involuntário ou invalidez total temporária, o seguro cobre até três parcelas do financiamento. Se o cliente vier a falecer ou sofrer invalidez permanente, a seguradora quita totalmente a dívida.

Para controlar e automatizar o fluxo operacional, a seguradora montou uma equipe de Tecnologia da Informação (TI) que desenvolveu internamente um software batizado de Life. Esse programa foi criado logo no início da operação da seguradora no Brasil, portanto possuí quase a mesma idade da empresa.

Estima-se que o Life tenha mais de cem mil linhas em seu código fonte. Esse software vem sendo alterado no decorrer de dez anos, para correção de erros, criação de novas funcionalidades, alterações devido ao surgimento de novos

produtos e adaptações às leis. Muitas dessas alterações tiveram a documentação extraviada ou nem chegaram a ser documentadas.

Para ilustrar o quanto essa empresa cresceu e mudou no decorrer desses dez anos, no início de sua operação contava com menos de dez funcionários e possuía um faturamento ínfimo comparado aos mais de cem milhões de reais obtidos após uma década e os quase quatrocentos funcionários que possui atualmente.

4.2 CONCEITOS DE SEGURO

Conforme mencionado anteriormente, para um melhor entendimento do estudo de caso, alguns conceitos do ramo de seguro precisam ser esclarecidos. No entanto, deve ser salientado que o universo de seguros é muito amplo e que será abordada somente uma pequena parte, relacionada aos principais impactos tratados nesse trabalho.

O primeiro conceito abordado é o de risco, que consiste no evento incerto, em que seu acontecimento independe da vontade tanto do segurado como da seguradora, e no qual o segurado deseja se precaver.

Caso esse evento ocorra, diz-se que ocorreu o sinistro, e a seguradora deve indenizar o segurado, de acordo com as regras previamente estabelecidas.

A cobertura estabelece os valores que a seguradora deve pagar ao segurado, caso ocorra o sinistro.

O prêmio é a importância que o segurado paga à seguradora, em troca da transferência do risco que está exposto. Em outras palavras, é o valor que o segurado paga para adquirir o seguro.

A apólice é a responsável por conter todos os parâmetros relacionados ao seguro comercializado. Na apólice constam os riscos, as coberturas, qual a empresa parceira que irá vender o seguro, a comissão que esta receberá por venda e qual o valor do prêmio. Diz-se que a apólice é a definição do produto que a seguradora irá comercializar.

Quando a pessoa adquire o seguro, a seguradora emite um certificado, que consta as informações como os dados pessoais do segurado, o número da apólice e

os riscos cobertos. Em outras palavras, cada certificado corresponde a um seguro existente.

4.3 ESTRUTURAÇÃO DO PROBLEMA E DEFINIÇÃO DOS OBJETIVOS

O Life foi criado para atender as necessidades da empresa, sendo suas principais funcionalidades o cadastro de apólices, visualização e edição das existentes, acompanhamento dos certificados, abertura e análise de sinistros e a geração de relatórios.

No entanto, esse programa possui alguns importantes problemas. Quando esse software foi concebido, a empresa era pequena, possuía poucos produtos e certificados, por isso foi criado visando atender as necessidades daquele momento. No decorrer dos anos, a quantidade de empresas parceiras, novos produtos e seguros vendidos aumentou vertiginosamente.

Um dos principais problemas que pode ser citado é o tempo necessário para implantar um novo produto. Sempre que a seguradora, juntamente com uma empresa parceira, criam um novo tipo de seguro, há a necessidade de se alterar o código fonte do Life, adaptando-o a esse novo produto. Isso implica no acionamento da equipe de TI, que irá fazer o levantamento e análise de requisitos, efetuará as alterações no código fonte do Life e encaminhará a nova versão do software para a homologação. Dependendo da complexidade do produto criado, esse fluxo pode levar até alguns meses para ser finalizado e ser possível o início das vendas. Com o mercado competitivo existente, esse tempo pode ser fatal diante de um concorrente mais ágil, podendo até mesmo resultar na perda da empresa parceira.

Outro problema apontado é o fato do Life ter sido concebido em uma época que havia pouca quantidade de dados para serem trabalhados e não se previa o grande aumento que sofreria. Isso resultou em um sistema com problemas de desempenho. Algumas funções do sistema demoram algumas horas para serem executadas e bloqueiam outras funcionalidades, impedindo os demais usuários de continuarem utilizando o programa.

Obviamente que existem outros problemas no Life. Devido a esse programa ser um arquivo executado no computador de cada estação de trabalho, existe a

necessidade de reinstalá-lo em todas as máquinas dos usuários a cada nova versão criada. Outro problema que pode ser citado é o fato de ser escrito em uma linguagem de programação obsoleta. A linguagem Visual Basic 6 (VB6) não possui mais suporte de seu fornecedor e nem o lançamento de novas ferramentas ou qualquer outro tipo de novidade que a empresa poderia se beneficiar.

Com os principais problemas identificados, a empresa passou a considerar a alternativa de migrar o sistema legado. Assim, iniciou-se o projeto de migração, com a missão de desenvolver um novo sistema, que eliminasse os problemas do software atual, mas continuasse atendendo as necessidades da empresa.

Nos objetivos da migração, foi definido que o novo sistema deveria ser construído em uma linguagem atual e que fosse flexível o bastante para que a criação de novos produtos necessitasse de um curto espaço de tempo, sem a necessidade de se alterar o código fonte. Também deveria ser desenvolvido para web, possibilitando a ampliação das máquinas que poderiam utilizá-lo e facilitando a publicação de novas versões.

Outro ponto importante definido é de que o novo sistema deverá ser único em todas as seguradoras pertencentes ao grupo, da América Latina. Atualmente, cada empresa do grupo, dessa região, utiliza sistemas próprios e possuem suas equipes para mantê-los funcionando.

4.4 ESTUDO DE VIABILIDADE

Para um correto estudo de viabilidade, é necessário primeiramente esclarecer os requisitos de negócios. A seguradora em estudo tem como objetivo seu crescimento através de duas linhas. A primeira consiste na conquista de novas empresas parceiras, que passariam a oferecer os seguros a seus clientes. A segunda linha é o aumento na diversificação dos tipos de seguros oferecidos para as empresas parceiras. Com isso, aumenta-se a possibilidade de venda, pois os clientes dessas empresas possuem uma diversidade cada vez maior de tipos de seguros que podem contratar.

Portanto, é esperada como vantagens do software que substituirá o Life, a possibilidade de inserir novas empresas parceiras e a criação de novos tipos de

seguros de uma maneira ágil. Também deve estar preparado para um grande aumento da quantidade de registros em sua base de dados, sem que isso resulte em uma queda significativa no desempenho do sistema como um todo.

Outra vantagem esperada com a migração é a padronização do software utilizado pelas seguradoras do grupo na América Latina. Haveria somente uma equipe responsável pela manutenção desse programa, facilitando a troca de conhecimento e experiências entre as empresas. Nos casos de diferenças entre leis e mercados que exigissem funcionalidades particulares, cada país deverá manter uma equipe de TI, que desenvolverão módulos a serem acoplados ao novo sistema.

Em relação à tecnologia, o novo sistema deverá ser desenvolvido em linguagem Java, que é considerada atual e já é empregado pela empresa. Além disso, a linguagem Java é multiplataforma, ou seja, pode ser executado em qualquer sistema operacional, sem a necessidade de alterações no código fonte.

No entanto, como desvantagem da substituição do Life por um novo sistema espera-se a ocorrência de algumas dificuldades, que podem se tornar sérios problemas caso não sejam tratadas corretamente. A primeira dificuldade está relacionada ao fato que o novo sistema pode apresentar erros ou problemas desconhecidos, que não foram detectados durante a fase de testes, e podem ocasionar impactos nas operações da empresa. Além disso, deve ser considerada a necessidade de um período de adaptação e treinamento dos usuários.

Como alternativa, pode-se manter o sistema atual. Mas, para que os problemas identificados sejam minimizados, seria necessário analisar toda a estrutura do banco de dados utilizada pelo Life, para adaptá-lo a enorme quantidade de informações armazenadas, com o intuito de melhorar o desempenho e reduzir os bloqueios.

Também seria necessário encontrar uma forma para tornar mais ágil à entrada de novas empresas parceiras e a criação de novos produtos, pois caso contrário, espera-se um aumento crescente do risco para o negócio, devido ao grande período necessário atualmente para a implantação de novos negócios.

Outro problema existente nessa alternativa é o fato de que alterações no código do Life ocasionam efeitos colaterais inesperados. Em outras palavras, devido à falta de uma documentação atualizada, a correção de um erro ou a criação de uma nova funcionalidade pode gerar erros em diversos lugares no sistema.

Além disso, outro problema que tende a agravar, caso o sistema legado seja mantido, está relacionado ao fato do Life ser escrito em uma linguagem defasada. Nota-se um crescente aumento na dificuldade de se encontrar profissionais que ainda programem em VB6. Isso resulta em aumento na média dos salários e uma maior demora para reposição de recursos.

Portanto, com o novo sistema espera-se um aumento na competitividade da empresa, devido a uma maior agilidade na criação de novos produtos e a melhor disseminação de conhecimento entre as empresas localizadas na América Latina. Também se estima uma redução nos custos, em virtude da diminuição das equipes de TI nas seguradoras desses países.

Com isso, pode-se concluir que com o novo sistema atendendo as necessidades listadas anteriormente, trará vantagens importantes em relação ao sistema atual e contribuirá para os objetivos da organização.

4.5 MAPEAMENTO DO SISTEMA LEGADO

Conforme descrito no modelo proposto, o sistema legado é constituído de diversos componentes que se relacionam entre si. Na figura 13 é ilustrado os componentes do sistema legado da seguradora.

Cada componente foi detalhado e feito a análise se a substituição do Life, por um novo programa, acarretará na necessidade de alterações.

O hardware do sistema legado é constituído dos computadores dos usuários, do servidor de banco de dados e o servidor de aplicação, utilizado pelo programa Upload, descrito posteriormente. O Life e a maioria dos demais softwares de aplicação são instalados e executados nas máquinas dos usuários.

Tanto os computadores dos usuários como o servidor de banco de dados não precisarão ser modificados ou substituídos devido à migração. No entanto, será necessário o upgrade ou a aquisição de um novo servidor de aplicação, tendo em vista que o novo programa será web.

Em relação aos softwares de apoio, devido ao fato da linguagem Visual Basic pertencer a Microsoft, esses softwares são basicamente dessa empresa. O sistema

operacional, tanto das máquinas dos usuários como dos servidores, é o Windows. O compilador do Life é o do VB6.

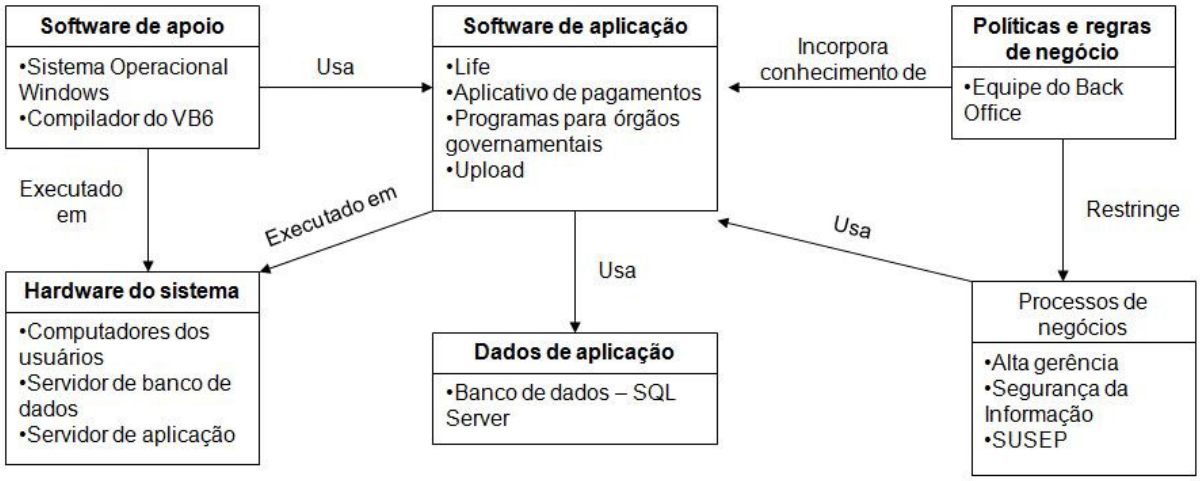

Figura 12 - Componentes do sistema legado da seguradora

O novo sistema, por ser multiplataforma, não exigirá a alteração do sistema operacional de nenhum computador da empresa. O compilador do VB6 não será mais necessário, sendo substituído pelo compilador do Java.

Os softwares de aplicação são constituídos além do próprio Life, do aplicativo responsável por efetuar os pagamentos às empresas parceiras e aos segurados, alguns programas para atendimento de exigências de órgãos governamentais e o software chamado de Upload, também desenvolvido internamente, mas em linguagem Java, e que efetua a troca de arquivos entre a seguradora e as empresas parceiras.

O aplicativo de pagamentos não necessitará de alterações, devido à migração do Life, pois toda a comunicação com esse programa é feito através de uma interface, que o novo sistema deve utilizar. Já os programas para atendimento de exigências de órgãos governamentais e o Upload acessam diretamente a base de dados do Life, sendo necessário readaptá-los para a estrutura da base de dados do novo sistema.

Os dados de aplicação são guardados em um servidor, através de um banco de dados relacional. O banco de dados adotado foi o SQL Server, também da Microsoft.

Um novo banco de dados será necessário, atendendo as exigências do programa que irá substituir o Life.

Os processos de negócio relacionados ao Life pertencem basicamente à equipe do Back Office. Essa equipe é responsável pelos cadastramentos dos produtos, das apólices, análise dos sinistros e geração de relatórios, sendo todas essas atividades feitas através do Life.

Esses processos precisarão ser adaptados ao modo como o novo software trabalhará. Treinamentos serão necessários aos usuários dessa equipe.

As políticas e regras de negócio são definidas pela alta gerência, pela área de segurança da informação e pela Superintendência Nacional de Seguros Privados (SUSEP), que é o órgão vinculado ao Ministério da Fazenda e possui a missão de regular, supervisionar, fiscalizar e incentivar as atividades de seguros. Nenhuma das políticas ou regras será alterada devido à migração.

4.6 ENGENHARIA REVERSA

O software Life é um programa complexo com diversas funcionalidades construídas para atender as mais variadas necessidades da empresa em análise, no ramo de seguros. Infelizmente, não é possível descrever todas essas funcionalidades no presente trabalho. Como o objetivo é a aplicação do modelo no cenário descrito, a engenharia reversa será empregada em duas importantes funções do Life.

A primeira função é o cadastro de uma nova apólice. Essa função foi escolhida para análise no presente trabalho, pois será a responsável por possibilitar, no novo sistema, a maior agilidade na criação de novos produtos na seguradora.

A segunda função a ser estudada é a análise de sinistro, devido ao fato de ser uma funcionalidade essencial no ramo de seguros e caso não opere corretamente no novo sistema, pode gerar grande insatisfação com os segurados e consequentemente ocasionar inúmeros processos judiciais.

Para cada uma das funções foi construído o diagrama de caso de uso e posteriormente o diagrama de objetos.

Na figura 14 é ilustrado o caso de uso do cadastro de uma nova apólice. Esse cadastro é necessário quando a seguradora fecha um acordo comercial com uma empresa parceira, para que esta inicie a venda de um novo produto de seguros.

Nesse caso, a área do Back Office recebe a documentação com todas as informações para a criação da nova apólice. O significado dessas informações pode ser mais bem compreendido na explicação do diagrama de objetos.

Como parte do processo de criação da apólice, na figura 14 também foi incluída o caso de uso ativar apólice. A ativação da apólice foi criada como forma de validação. Quando uma nova apólice é cadastrada, esta fica com o estado de pendente. Nesse estado, nenhum certificado dessa apólice pode ser comercializado, até que seja feita a ativação.

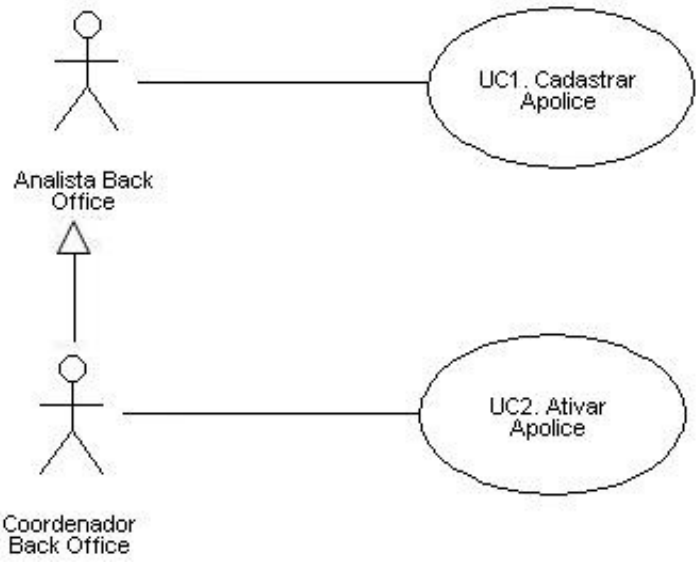

Figura 13 – Caso de uso do cadastro de apólice

Nos quadros 1 e 2 são descritos os casos de uso da figura 14.

UC1. Cadastrar Apolice	
Descrição:	Este caso de uso descreve a forma de cadastrar uma nova apólice no sistema.
Atores:	Analista do Back Office
Pré-condição:	Analista autenticado no sistema
Fluxo principal:	1) O analista solicita o cadastramento de uma nova apólice 2) O sistema apresenta a lista das empresas parceiras cadastradas. 3) O analista seleciona a empresa parceira correspondente à nova apólice. 4) O sistema solicita o preenchimento das seguintes informações: a. Descritivo da apólice b. Valor do prêmio

	c. Valor de comissão a ser pago à empresa parceira d. Datas de início e fim de vigência da apólice 5) O analista preenche as informações solicitadas. 6) O sistema apresenta a lista de riscos. 7) O analista seleciona os riscos cobertos pela apólice. 8) O sistema solicita as seguintes informações, para cada risco: • Documentos obrigatórios • Valor da cobertura 9) O analista preenche as informações relativas aos riscos. 10) O sistema confirma o cadastramento.
Pós-condição:	Não há.
Fluxo alternativo:	(Passo 6) O valor de comissão é maior ou igual ao valor do prêmio. O sistema emite uma mensagem de erro e volta ao passo 4. (Passo 10) A data de fim de vigência é menor que a data de início de vigência. O sistema emite uma mensagem de erro e volta ao passo 4. (Passo 10) O analista deixou uma das informações em branco. O sistema emite uma mensagem de erro e volta ao passo 4.
Requisitos especiais:	Não há.

Quadro 1 – Descrição do caso de uso cadastrar apólice

UC2. Ativar Apólice	
Descrição:	Este caso de uso descreve a forma de ativar uma apólice nova.
Atores:	Coordenador do Back Office
Pré-condição:	Coordenador autenticado no sistema.
Fluxo principal:	1) O coordenador solicita a ativação de uma apólice. 2) O sistema lista as empresas parceiras cadastradas. 3) O coordenador seleciona a empresa que possui uma apólice a ser ativada. 4) O sistema lista todas as apólices dessa empresa parceira, com o estado de pendente. 5) O coordenador seleciona a apólice a ser ativada. 6) O sistema exibe todas as informações cadastradas nessa apólice. 7) O coordenador ativa à apólice. 8) O sistema confirma a ativação.
Pós-condição:	Apólice ativada.
Fluxo alternativo:	(Passo 7) O coordenador verificou que o cadastro está incorreto. O estado de pendente é mantido.

Requisitos especiais:	Não há.

Quadro 2 – Descrição do caso de uso ativar apólice

Após ser realizada uma análise tanto no código fonte do Life como na estrutura das tabelas do banco de dados, foi possível a construção do diagrama de objetos da apólice, que consta na figura 15. Para que não ficasse confuso devido ao excesso de informações, o diagrama foi simplificado.

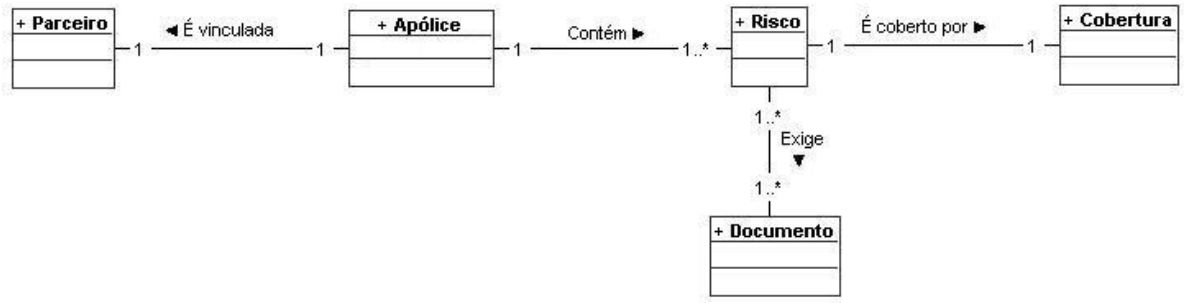

Figura 14 – Diagrama de objetos da apólice

Através do diagrama de objetos da figura 15 é possível visualizar a relação de multiplicidade de uma apólice cadastrada. Toda apólice possui um parceiro vinculado, que será a empresa que irá comercializar o seguro, também contém pelo menos um risco e cada risco possuí uma cobertura.

Na apólice também é configurado os documentos obrigatórios para cada risco. Esse parâmetro indica quais documentos o segurado deve apresentar quando solicita a abertura de um sinistro, para que seja comprovada a ocorrência do mesmo.

Essa é a estrutura básica que o analista do Back Office deve ter em mente para cadastrar uma apólice.

A outra funcionalidade a ser descrita é a de análise de sinistro. Há duas formas de um sinistro ser aberto. A primeira é o segurado comunicar diretamente a empresa parceira que houve a ocorrência de um sinistro, que por sua vez envia regularmente arquivos, informando a seguradora sobre esses fatos. A segunda forma é o próprio segurado ligar para o atendimento da seguradora e solicitar a abertura do sinistro. Em ambos os casos, os sinistros são analisados pela equipe do Back Office, através do Life.

O diagrama de caso de uso da análise de sinistro consta na figura 16, e contempla todo o processo de análise do sinistro após a sua abertura até a

aprovação do pagamento. Nesse diagrama, além do caso de uso analisar sinistro, que refere-se à primeira análise feita no sinistro, há o caso de uso digitalizar documento, no qual os documentos que compravam a ocorrência do sinistro, enviados pelo segurado, são digitalizados, ou seja, os documentos físicos são inseridos em um escâner e convertidos para formato de arquivos eletrônicos. Com os arquivos digitalizados, estes são vinculados ao sinistro. O caso de uso aprovar pagamento é relativo à forma de validação criada, para garantir que os valores a serem pagos estão corretos.

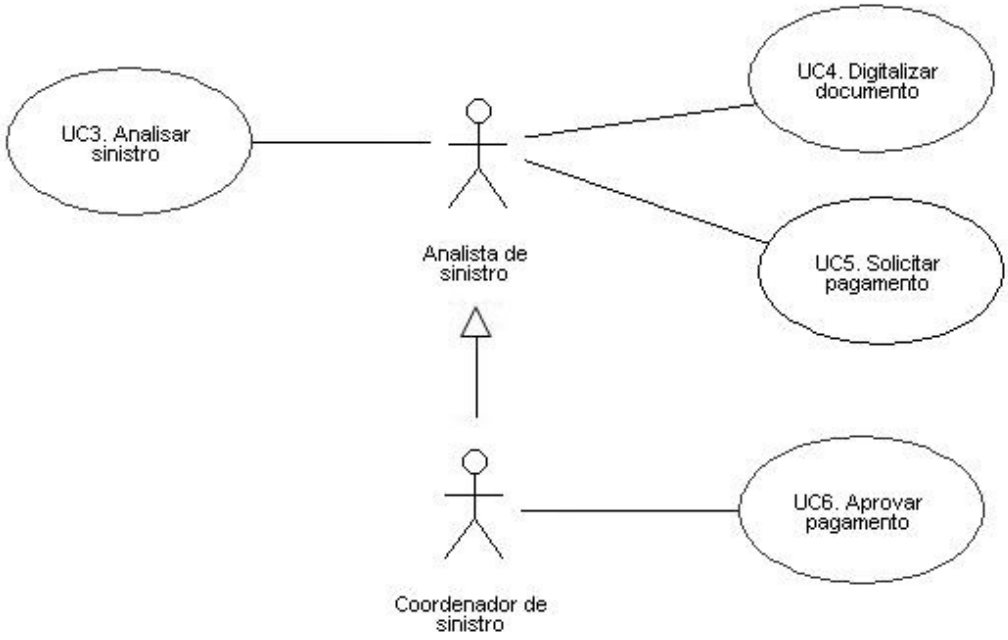

Figura 15 – Diagrama de caso de uso da análise de sinistro

As descrições de cada caso de uso são feitas nos quadros a seguir.

UC3. Analisar sinistro	
Descrição:	Este caso de uso descreve a analise inicial de um novo sinistro aberto.
Atores:	Analista de sinistro
Pré-condição:	Analista autenticado no sistema
Fluxo principal:	1) O analista solicita a análise de sinistro não analisado. 2) O sistema apresenta a lista de sinistros ainda não analisados. 3) O analista seleciona o sinistro a ser analisado. 4) O sistema exibe os dados relativos ao sinistro, com as seguintes informações: a. Nome do segurado. b. Período de vigência do certificado. c. Risco ocorrido. d. Valor da cobertura para o risco ocorrido.

	e. Documentos obrigatórios. 5) O analista solicita ao sistema a geração da carta a ser enviada ao segurado, solicitando a documentação. 6) O sistema gera a carta.
Pós-condição:	Não há.
Fluxo alternativo:	(Passo 6) O sinistro ocorreu fora do período de vigência do certificado. O sistema emite uma mensagem de erro e gera a carta de declínio, explicando ao segurado que o sinistro será recusado.
Requisitos especiais:	Não há.

Quadro 3 – Descrição do caso de uso analisar sinistro

UC4. Digitalizar documento	
Descrição:	Este caso de uso descreve a forma de digitalizar um documento de um sinistro.
Atores:	Analista de sinistro
Pré-condição:	Analista autenticado no sistema
Fluxo principal:	1) O analista solicita a digitalização de documento. 2) O sistema solicita a identificação do sinistro. 3) O analista informa a identificação do sinistro. 4) O sistema apresenta a lista dos documentos obrigatórios a esse sinistro. 5) O analista seleciona o documento a ser digitalizado. 6) O sistema solicita a digitalização do documento. 7) O analista digitaliza o documento. 8) O sistema confirma a digitalização.
Pós-condição:	Não há.
Fluxo alternativo:	(Passo 4) O sistema não encontrou o sinistro através da identificação. O sistema emite uma mensagem de erro e retorna ao passo 2.
Requisitos especiais:	Não há.

Quadro 4 – Descrição do caso de uso digitalizar documento.

UC5. Solicitar pagamento	
Descrição:	Este caso de uso descreve a forma de solicitar o pagamento para um sinistro
Atores:	Analista de sinistro
Pré-condição:	Analista autenticado no sistema
Fluxo principal:	1) O analista solicita o pagamento de sinistro. 2) O sistema apresenta a lista com os sinistros que tiveram os documentos obrigatórios já digitalizados. 3) O analista seleciona o sinistro que será solicitado o pagamento.

	4) O sistema exibe os dados do pagamento do sinistro, com as seguintes informações: a. Valor da cobertura b. Dados bancários do favorecido. 5) O analista solicita o pagamento ao favorecido. 6) O sistema solicita a data do pagamento. 7) O analista informa a data do pagamento. 8) O sistema confirma a solicitação.
Pós-condição:	Não há.
Fluxo alternativo:	(Passo 6) Os dados bancários não estão preenchidos. O sistema emite uma mensagem de erro e retorna ao passo 4.
Requisitos especiais:	Não há.

Quadro 5 – Descrição do caso de uso solicitar pagamento.

UC6. Aprovar pagamento	
Descrição:	Este caso de uso descreve a forma de aprovar os pagamentos de um sinistro.
Atores:	Coordenador de sinistro
Pré-condição:	Coordenador autenticado no sistema
Fluxo principal:	1) O coordenador solicita a aprovação de pagamento 2) O sistema apresenta a lista dos sinistros já analisados e com pagamentos. 3) O coordenador seleciona um dos sinistros. 4) O sistema apresenta os dados do pagamento do sinistro, com as seguintes informações: a. Valor a ser pago. b. Data do pagamento. c. Dados bancários do favorecido. 5) O coordenador aprova o pagamento. 6) O sistema confirma a aprovação e envia o pagamento para o setor responsável por fazê-lo.
Pós-condição:	Não há.
Fluxo alternativo:	Não há.
Requisitos especiais:	Não há.

Quadro 6 – Descrição do caso de uso aprovar pagamento

O diagrama de objetos simplificado de um sinistro é ilustrado na figura 17. Nesse diagrama é possível verificar que cada sinistro foi aberto por um certificado, ou seja, o sinistro só pode ser aberto devido à existência do certificado. Também

nota-se que todo sinistro precisa de pelo menos um documento que comprove a sua ocorrência.

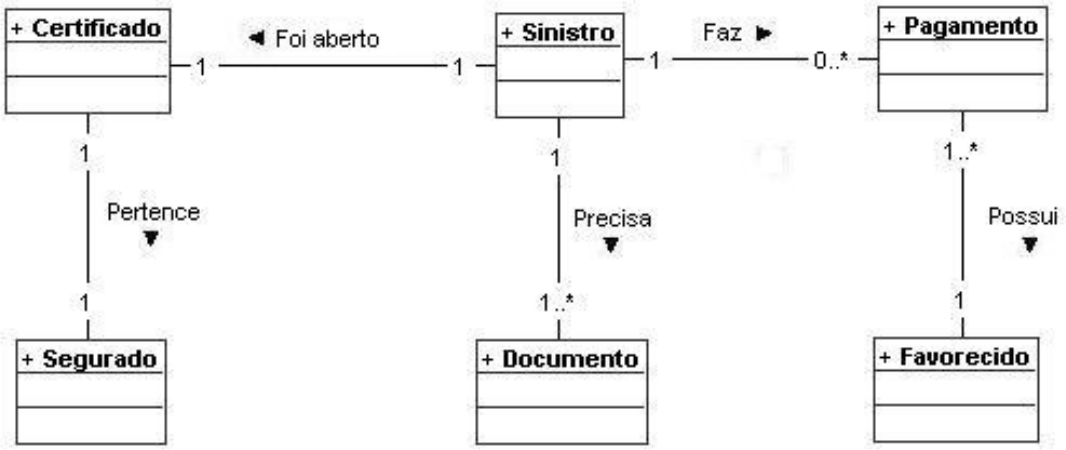

Figura 16 – Diagrama de objetos de um sinistro

Através do diagrama também é possível verificar que o sinistro pode fazer nenhum pagamento. Esse caso ocorre quando o sinistro é rejeitado, devido a alguma regra de negócio que impede o pagamento. Já os casos de serem feitos mais de um pagamento, são os casos de desemprego, onde a seguradora deve pagar até certa quantidade de parcelas do financiamento, enquanto o segurado estiver desempregado. A quantidade máxima de parcelas que a seguradora deve pagar é configurada na cobertura da apólice. Todo pagamento deve ter um favorecido, ou seja, quem irá recebê-lo.

4.7 ANÁLISE DE REQUISITOS

Depois de feito o mapeamento do sistema legado e a engenharia reversa têm-se uma visão de como o Life interage com os outros componentes e como as funcionalidades são utilizadas. Esse conhecimento deve facilitar um melhor entendimento dos requisitos feitos para o novo programa.

Na análise de requisitos, foram consultados os gestores da área de TI e os funcionários da equipe do Back Office.

Essa subseção trata dos requisitos das duas funcionalidades descritas na engenharia reversa. Esses requisitos podem ser separados de acordo com os

pontos de vista. Os requisitos de melhoria para o cadastro da apólice é um requisito do ponto de vista indireto, pois foi solicitado pela alta gerência da equipe de TI. Já os requisitos, para a funcionalidade de análise de sinistros, pertencem ao ponto de vista de interação.

A melhoria solicitada para o cadastro da apólice é a possibilidade de uma maior flexibilidade na configuração dos seus parâmetros. Atualmente há alterações que só são possíveis alterando o código fonte do Life. A idéia é de que o analista do Back Office possa criar ou alterar um parâmetro, através de fórmulas, sem a necessidade de se modificar o código fonte.

Um exemplo que pode ser feito para ilustrar esse cenário é em relação ao valor do prêmio, configurado na apólice. No Life, é possível configurar o prêmio como um valor fixo ou o percentual do valor do bem que está sendo segurado. Mas se for criado um produto em que o prêmio é a soma de um valor fixo mais um percentual do valor do bem segurado, será necessário a alteração no código para que o Life passe a aceitar também essa configuração. No novo sistema, o analista do Back Office que for cadastrar essa apólice, deve conseguir configurar o prêmio dessa maneira através do próprio programa.

Na figura 18 é ilustrado o caso de uso do cadastro da apólice, com uma nova funcionalidade, a cópia de uma apólice existente. Além da maior flexibilidade, o novo sistema deve permitir a criação de uma nova apólice a partir de outra existente. Com isso, espera-se uma maior agilidade na criação de apólices semelhantes.

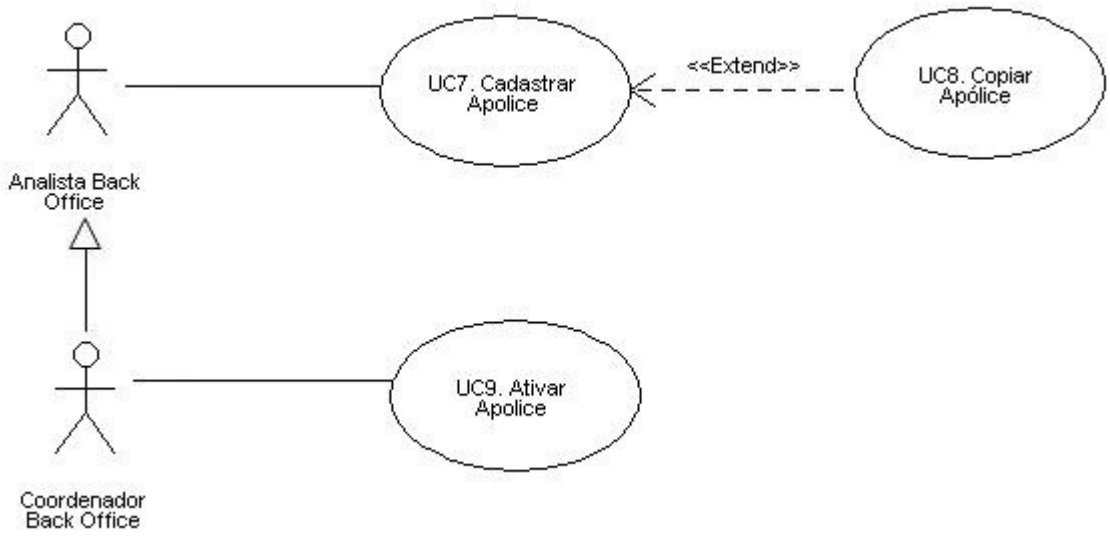

Figura 17 – Caso de uso do Cadastro de Apólice

Nos quadros a seguir são feitas as descrições dos casos de uso cadastrar apólice e copiar apólice. Para o caso de uso ativar apólice não houve solicitação de alterações ou melhorias, portanto deve ser considerada a mesma descrição feita durante a engenharia reversa, ou seja, a descrição do caso de uso UC9 é igual ao UC2.

UC7. Cadastrar Apólice	
Descrição:	Este caso de uso descreve a forma de cadastrar uma nova apólice no novo sistema.
Atores:	Analista do Back Office
Pré-condição:	Analista autenticado no sistema
Fluxo principal:	1) O analista solicita o cadastramento de uma nova apólice 2) O sistema apresenta a lista das empresas parceiras cadastradas. 3) O analista seleciona a empresa parceira correspondente à nova apólice. 4) O sistema solicita o preenchimento das seguintes informações: e. Descritivo da apólice f. Datas de início e fim de vigência da apólice 5) O analista preenche as informações solicitadas. 6) O sistema solicita o preenchimento das fórmulas que fornecem as seguintes informações: g. Valor do prêmio h. Valor de comissão a ser pago à empresa parceira 7) O analista preenche as fórmulas solicitadas. 8) O sistema apresenta a lista de riscos cadastrados. 9) O analista seleciona os riscos cobertos pela apólice. 10) O sistema solicita as seguintes informações, para cada risco: • Documentos obrigatórios • Fórmula que define o valor da cobertura 11) O analista preenche as informações relativas aos riscos. 12) O sistema confirma o cadastramento.
Pós-condição:	Não há.
Fluxo alternativo:	(Passo 6) A data de fim de vigência é menor que a data de início de vigência. O sistema emite uma mensagem de erro e volta ao passo 4. (Passo 8) O valor da comissão é maior ou igual ao valor do prêmio. O sistema emite uma mensagem de erro e retorna ao passo 6. (Passo 12) O analista deixou uma das informações em

	branco. O sistema emite uma mensagem de erro e volta ao passo 4. (Passo 12) A apólice cadastrada é igual a uma já existente. O sistema emite uma mensagem de erro e volta ao passo 4.
Requisitos especiais:	Não há.

Quadro 7 – Descrição do caso de uso cadastrar apólice.

UC8. Copiar apólice	
Descrição:	Este caso de uso descreve a forma de solicitar a copia das configurações de uma apólice, na criação de uma nova.
Atores:	Analista do Back Office
Pré-condição:	Analista está no cadastro de uma nova apólice.
Fluxo principal:	1) O analista solicita a copia dos parâmetros de uma apólice existente. 2) O sistema apresenta a lista com as empresas parceiras. 3) O analista seleciona a empresa parceira, que possui a apólice a ser copiada. 4) O sistema lista as apólices vinculadas a essa empresa parceira. 5) O analista seleciona a apólice a ser copiada. 6) O sistema carrega nos respectivos parâmetros da nova apólice, as seguintes informações da apólice selecionada. a. Riscos cobertos. b. Documentos obrigatórios. c. Fórmula que define o valor da cobertura.
Pós-condição:	Sistema retorna ao passo 2 do caso de uso Cadastrar Apólice.
Fluxo alternativo:	(Passo 4) A empresa parceira selecionada não possui nenhuma apólice cadastrada. O sistema emite uma mensagem de erro e retorna ao passo 2.
Requisitos especiais:	Não há.

Quadro 8 – Descrição do caso de uso copiar apólice.

Vale ressaltar que a apesar de uma nova apólice ter sido cadastrada utilizando a funcionalidade de copiar apólice, as duas apólices não precisam estar vinculadas a mesma empresa parceira, o que justifica o fato do término da funcionalidade de cópia da apólice, se retornar ao passo em que o analista do Back Office deve selecionar a empresa parceira da nova apólice.

Para a funcionalidade de análise de sinistro, não houve alteração tanto no diagrama de caso de uso como nas descrições levantadas na engenharia reversa. No entanto, foram solicitadas funções que visam auxiliar na análise pelo Back Office.

No sistema atual, quando é necessário o envio de carta solicitando ao segurado a documentação obrigatória do sinistro, é impresso a carta e enviada pelo correio. O segurado por sua vez, também envia a cópia dos documentos fisicamente. Visando tornar mais ágil esse processo, foi requisitada a opção de envido da carta de solicitação de documentos, por e-mail. O segurado também poderia enviar a documentação através de arquivos eletrônicos, pela internet.

Outro requisito feito pela equipe do Back Office é para os casos em que os segurados precisam ser examinados por um médico. No Life, é solicitado como um documento obrigatório, onde o segurado pode enviar a cópia de um atestado falso. Para o novo sistema, o segurado deveria ser encaminhado para um médico cadastrado pela seguradora, para ser examinado. O médico teria acesso, através da internet, a um aplicativo em que inserisse seu parecer e enviasse diretamente à seguradora, diminuindo o risco de fraudes.

5 CONCLUSÕES

Analisando os estudos feitos no presente trabalho, pode-se concluir que quando uma empresa estuda a hipótese de migrar seu sistema legado, deve ter claro todos os problemas que esse sistema apresenta. Esse esclarecimento mostra-se importante na definição dos objetivos que o novo sistema deve apresentar e auxilia no estudo se a substituição do sistema existente é viável ou não.

Também foi possível verificar que no sistema legado existem informações importantes ao negócio, como funcionalidades e regras de negócio, que podem ser o diferencial da empresa perante seus concorrentes. Descartá-los ou confiar somente no conhecimento das pessoas envolvidas com os processos pode resultar em prejuízos como a execução de procedimentos de forma equivocada ou até mesmo o não atendimento a leis.

Além disso, mesmo que o objetivo seja somente migrar um único programa, esse software provavelmente pertence a um sistema que será impactado com a sua substituição. Conhecer previamente todo esse sistema e saber quais partes podem exigir algum tipo de alteração, diminuirão as chances de ocorrer problemas que serão somente detectados quando o novo software entrar em operação e precisar interagir com todos os componentes.

Conhecer também a forma como o legado trabalha auxilia no levantamento dos requisitos para o novo sistema. Primeiramente porque os engenheiros de software precisarão entender o contexto de como o legado é utilizado e a importância das funcionalidades existentes. Outro motivo porque provavelmente todas as pessoas que solicitarão os requisitos conhecem o legado e farão seus pedidos baseados em como gostariam que o legado trabalhasse.

O modelo proposto sugere que o processo de mapeamento do sistema legado, a engenharia reversa e a análise de requisitos sejam trabalhadas como um ciclo iterativo principalmente para melhorar o entendimento do legado, através de um melhor detalhamento.

Mas o acompanhamento do projeto de migração da empresa analisada mostrou que projetos desse tipo podem demorar anos para serem finalizados. Durante esse período, a empresa precisa continuar crescendo e se adaptando as

novas necessidades. Tal fato resulta em alterações no legado, que devem ser replicadas ao novo sistema. O ciclo iterativo também poderia auxiliar no processo de manter a documentação atualizada, evitando que o novo sistema seja construído baseado em um cenário obsoleto.

Portanto, devido à grande importância que os sistemas legados normalmente possuem para as empresas, nota-se que o emprego de um modelo para auxiliar na migração, pode evitar sérios problemas como grandes atrasos na construção do novo sistema e também reduzir os riscos de se ter um usuário descontente após a migração, além de diminuir as chances do novo sistema não atender as necessidades da empresa.

5.2 TRABALHOS FUTUROS

Como trabalhos futuros, que podem vir a complementar esse trabalho e, consequentemente, o assunto sobre migração de sistemas legados, recomenda-se o estudo sobre outras técnicas e ferramentas para o levantamento das informações dos sistemas legados. Outro assunto que poderia ser abordado é a análise sobre a fase de execução de um projeto de migração.

REFERÊNCIAS

BRAUDE E. **Projeto de software:** da programação à arquitetura: uma abordagem baseada em java. São Paulo: Bookman, 2004.

BOOCH G.; RUMBAUGH J.; JACOBSON I. **UML guia do usuário.** 14 ed. Rio de Janeiro: Campus, 2000. 472 p.

CAMPOS, P. **Engenharia de requisitos: Iniciando um projeto de ER.** Portugal: Universidade da Madeira. 2007. Disponível em: <http://dme.uma.pt/edu/er/slides/ER-IniciandoProjectoER.pdf>. Acesso em: 11 out. 2009.

CHIARELLO M.; MAYER D. **Guia de especificação de casos de uso.** Paraná: CELEPAR, 2009.

EARLS A.B.; EMBURY S. M.; TURNER N. H. A method for the manual extraction of business rules from legacy source code**. BT Technology Journal,** v. 20, n. 4, Oct. 2002. Disponível em <http://resources.metapress.com/pdf-preview.axd?code=j1181787w771742x&size=largest> Acesso em 05 jan. 2010.

FERREIRA A. **Análise de ferramentas de modelagem UML gratuitas.** Rio Grande do Sul. Universidade Federal do Rio Grande do Sul. 2001. Disponível em: < http://www.inf.ufrgs.br/gppd/disc/cmp167/trabalhos/sem2001-1/T1/alex/>. Acesso em: 05 dez. 2009.

GANDIN, S. J. **Migração de sistemas legados.** 2003. 53 p. Dissertação (Mestrado) – Instituto de Informática, Universidade Federal do Rio Grande do Sul. Porto Alegre, 2003.

HORA B. V. **Revisão de crenças no gragmento universal da CTL usando verificação de modelos limitada.** São Paulo. Universidade de São Paulo. 2009. Disponível em <http://www.ime.usp.br/~cef/mac499-09/monografias/bruno-da-hora/monografia.pdf>. Acesso em: 24 dez. 2009.

KOBRYN C. et al. **UML essencial:** Um breve guia para a linguagem-padrão de modelagem de objetos. 3 ed. São Paulo: Bookman, 2005.

KON F. **Uma visão geral de UML.** 2005, 65 p. IME, Universidade de São Paulo. Dísponível em: <http://www.ime.usp.br/~kon/presentations/UMLIntro.pdf>. Acesso em 07 jan. 2010.

LINS, F. A. A. **Fases e ciclo de vida de um projeto.** Pernambuco: Universidade Federal de Pernambuco, 2007. 16 p. Disponível em: <http://www.nti.ufrpe.br/ fernandoaires/disciplinas/gestaoprojetos/Aulas/aula04.ppt>. Acesso em 01 nov. 2009.

PREVIDELLI, C. A. **Migração de sistemas legados para ambiente de suporte a projetos e operação**. 2001. 123 p. Dissertação (mestrado) – Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2005.

PROJECT MANAGEMENT INSTITUTE, **Um guia do conjunto de conhecimento em gerenciamento de projetos (Guia PMBOK)**. 3 ed. Pennylvania: 2004.

PRESSMAN, R. S. **Engenharia de software**. 6 ed. São Paulo: McGraw-Hill, 2006.

RECHIA E. L.; PENTEADO R. **Avaliação da aplicabilidade da linguagem de padrões de engenharia reversa de Demeyer a sistemas legados procedimentais.** 2002. Universidade Federal do Rio de Janeiro. Disponível em: < http://lens.cos.ufrj.br/sugarloafplop/2002/papers/SPA2.pdf>. Acesso em: 05 jan. 2010.

RICARTE I. L. M. **Programação orientada a objetos com C++.** Campinas: UNICAMP. 2001. 76 p. Apostila para disciplina de graduação do Departamento de Engenharia de Computação e Automação Industrial.

SAMPAIO M. C. **Unified modeling language.** Paraíba: Universidade Federal de Campina Grande. 2007. Material para a disciplina de Sistema de Informação II. Disponível em: <http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/index.htm>. Acesso em: 09 dez. 2009.

SANTANDER V. F. A.; CASTRO J. F. B. **Desenvolvendo use case a partir de modelagem organizacional.** Pernambuco: Universidade Federal de Pernambuco, 2004. 23 p.

SATZINGER J.; JACKSON R.; BURD S. **Systems analysis & design in a changing world.** 5 ed. Boston: Cengage, 2009. 686 p.

SCOTT, K. O processo unificado explicado explicado. São Paulo: Bookman: 2002.

SILVA A. C. **Unified Modeling Language (UML).** Maranhão: Universidade Federal do Maranhão, 2006. 21 p. Disponível em: <http://www.deinf.ufma.br/~acmo/MOO_Classe.pdf>. Acesso em: 07 jan. 2010.

SOMMERVILLE, I. **Engenharia de software.** 8. ed. São Paulo: Pearson Addison – Wesley, 2007.

SOUZA, C. **Modelagem de caso de uso:** diagrama de caso de uso. Paraná: Universidade Federal do Paraná, 2006. Disponível em: < http://www.ufpa.br/cdesouza/teaching/es/6-use-cases.pdf>. Acesso em: 29 dez. 2009.

VARGAS R. **Manual prático do plano de projeto.** 3. ed. Rio de Janeiro: Brasport, 2007.

VASCONCELOS A. P. V. **Uma abordagem de apoio à criação de arquiteturas de referência de domínio baseada na análise de sistemas legados.** 2007. 267 p. Tese (Doutorado) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.