

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciências da Computação

JOSÉ MAGNO DE FREITAS CARDOSO

**MODELAGEM DE REQUISITOS BASEADO EM CASOS DE USO PARA UM
SISTEMA LEGADO**

Belo Horizonte
2011

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciências da Computação
Especialização em Informática: Ênfase: Análise de Sistemas

**MODELAGEM DE REQUISISTO BASEADO EM
CASOS DE USO PARA UM SISTEMA LEGADO**

por

JOSÉ MAGNO DE FREITAS CARDOSO

Monografia de Final de Curso

Prof. Antônio Mendes Ribeiro
Orientador

Belo Horizonte
2011

JOSÉ MAGNO DE FREITAS CARDOSO

MODELAGEM DE REQUISITOS BASEADO EM CASOS DE USO PARA UM SISTEMA LEGADO

Monografia apresentada ao Curso de Especialização em Informática do Departamento de Ciências Exatas da Universidade Federal de Minas Gerais, como atividade da disciplina Desenvolvimento de Pesquisa e Projetos em Informática, ministrada pelo professor Antônio Mendes Ribeiro.

Área de concentração: *Análise de Sistemas*

Orientador(a): Prof. *Antônio Mendes Ribeiro*

Belo Horizonte
2011

RESUMO

Novos módulos, produtos e funcionalidades estão constantemente sendo especificados para sistemas de cartões de crédito implementados em mainframe. Formulários ou Templates próprios para especificações de requisitos são preenchidos utilizando descrições textuais formais. Apesar da utilização de modelos para requisitos de negócios e requisitos funcionais, o índice de erros tem sido elevado. Com frequência as especificações de requisitos não estão claras o suficiente para o desenvolvimento no prazo e com a qualidade esperada pelos demandantes. Não é a intenção deste trabalho propor mudança de metodologia (enfoque do desenvolvimento em “Cascata”) para as empresas do setor. Busca-se no entanto estudar e apresentar o Estudo de Caso com a utilização de novas técnicas, como a utilização de casos de uso, para melhorar o entendimento entre as partes envolvidas no que diz respeito a requisitos comportamentais, resultando em uma melhor qualidade no produto final. A Engenharia de software orientada a objetos se popularizou na década de 90 e nesta época Ivar Jacobson apresentou “Casos de Uso”, sua criação - elaborada 2 décadas antes, à comunidade de programação Orientada a Objeto conforme relatado por CockBurn (1999). A Engenharia de Software enfatiza a necessidade do levantamento de requisitos para o desenvolvimento de sistemas, evidenciada pela constatação de projetos com grande incidência de erros em termos de prazo, recursos e até funcionalidades que não atendem adequadamente aos clientes (SAMPAIO; PRIMO; MARTINHO, 2005). As técnicas de Orientação a objetos com a possibilidade de derivação de Casos de Uso a partir de requisitos motivam o trabalho de pesquisa da presente monografia, que deverá focar nas técnicas para modelagem de requisitos baseada em Cenários, Histórias de Usuário e Casos de Uso.

Palavras-chave: Casos de Uso, Cenário, Histórias de Usuário, Programação Orientada a Objeto

ABSTRACT

New modules, products and features are frequently being specified to credit card systems implemented in mainframe. Own forms or templates for requirements specifications are filled using formal textual descriptions. Although the use of models of business requirements and functional requirements, the error rate has been elevated. Often the requirements specifications are not clear enough for the development on time and with the quality expected by the stakeholders. It is not the intention of this work to propose change in methodology actually used (development focusing in “cascade”) by companies. However the intention is to study and present a case study that use new techniques such as Use Cases to improve understanding between every one involved in the work in relation to behavioral requirements, resulting in better product quality end. Engineering object-oriented software became popular in the 90s and at this time Ivar Jacobson presented “Use Cases”, his creation – designed 2 decades ago, to the community Object Oriented Programming as reported by CockBurn (1999). Software Engineering emphasizes the need for gathering requirements for system development, as evidenced by the finding projects with high rates of error in terms of time, resources and even features that not agree properly with customers necessity (SAMPAIO; PRIMO; MARTINHO, 2005). The techniques of object-orientation with the possibility of deriving Use Cases from requirements motivate the research of this monograph, which will focus on techniques for modeling requirements base on scenarios, User Stories and Use Cases.

Keywords: Use Cases, Scenario, User Stories, Object-oriented programming

LISTA DE FIGURAS

FIGURA 1 Casos de Uso permitindo a realização de objetivos	17
FIGURA 2 Formalidade, esforço de documentação e detalhes capturados	24
FIGURA 3 Mapas mentais para representar hierarquia de objetivos/requisitos	35
FIGURA 4 Casos de Uso marcados no nível do Usuário	37

LISTA DE QUADROS

QUADRO 1 Lista de Objetivos por Ator e Sistema	30
QUADRO 2 Hierarquia de Requisitos Concreto, nível do Usuário e com exceções	33

LISTA DE SIGLAS

RUP Rational Unified Process
UML Unified Modeling Language

SUMÁRIO

1 INTRODUÇÃO	10
2 LEVANTAMENTO DE REQUISITOS.....	11
2.1 HISTÓRIAS DE USUÁRIO	12
2.1.1 Características de uma boa História de Usuário	12
2.1.2 Os 3 C's das Histórias de Usuário	13
2.2 DESENVOLVIMENTO DE CASOS DE USO.....	15
2.2.1 Conceitos	15
2.2.2 Definindo e estruturando objetivos.....	16
2.2.3 Especificando com UML a partir de objetivos	17
2.2.4 Refinando objetivos	18
2.2.5 Priorizando objetivos	18
2.2.6 Escopo.....	19
2.2.7 Níveis de Casos de Uso	20
2.2.8 Formatos de Casos de Uso.....	20
2.3 CENÁRIOS.....	21
2.3.1 Cenário de sucesso principal.....	21
2.3.2 Extensões	22
3 ANÁLISE DAS TÉCNICAS APRESENTADAS	23
3.1 ANÁLISE COMPARATIVA DAS TÉCNICAS APRESENTADAS	23
4 ESTUDO DE CASO	25
4.1 DESCRIÇÃO DO PROBLEMA	25
5 CONCLUSÃO.....	39
REFERÊNCIAS	41

1 INTRODUÇÃO

É bastante comum as especificações de requisitos chegarem à equipe de desenvolvimento para produtos do setor de meios de pagamentos (mais especificamente cartões de crédito) com qualidade questionável. As demandas solicitadas pelas Administradoras de cartões são documentadas em formulário destinado a descrever os requisitos de negócios. Este documento serve como base para o levantamento dos requisitos funcionais. Às vezes utilizando-se de técnicas estruturadas ou mesmo sem a utilização de uma técnica específica, um documento com os requisitos funcionais levantados é elaborado e enviado para a equipe de desenvolvimento. É frequente depararmos com a situação em que o que é desenvolvido não representa exatamente o desejo dos Stakeholders. O retorno ao desenvolvimento para adequações e correções onera o processo, compromete os prazos acordados e leva à insatisfação dos demandantes. Vale lembrar que não é utilizada uma metodologia moderna com um processo iterativo e incremental.

Estudar e examinar as técnicas de modelagem de requisitos baseada em Cenários, histórias de usuário e casos de uso, buscando sedimentar conhecimentos para aplicar nas especificações de requisitos e desenvolvimento de demandas para a área de cartões de crédito em ambiente mainframe, complementando os recursos já utilizados, constituem a motivação deste trabalho.

Este trabalho tem como objetivos específicos apresentar as técnicas para o levantamento de requisitos, efetuar uma análise comparativa entre elas. Primeiramente os conceitos de História de Usuário, Casos de Uso, Cenários, objetivos, propriedades do domínio, requisitos e especificação são apresentados. Em seguida é descrito como definir e estruturar os objetivos. A partir dos objetivos é abordada a especificação com UML (elicitar contexto do sistema, definir objetivos do sistema, derivar requisitos, derivar Casos de Uso, derivar modelos UML). O trabalho prossegue com o Refinamento dos objetivos criando uma hierarquia de objetivos. Ainda tratando de Casos de Uso temos a definição do Escopo e os seus Níveis e Formatos. O próximo passo é a apresentação do conceito de Cenários para os Casos de Uso. Por fim, veremos um estudo de caso com a utilização das técnicas estudadas.

2 LEVANTAMENTO DE REQUISITOS

“Os produtos mais úteis são aqueles cujos desenvolvedores entenderam o que o produto deve fazer por seus usuários e como deve realizar seu objetivo” (Robertson, Robertson, 2006, p. 15). Ainda conforme os autores do livro “Mastering the Requirements Process”,

“Os requisitos corretos vem do entendimento de que trabalho que o produto se destina a apoiar. Somente quando você conhece os requisitos corretos você poderá projetar e construir o produto corretamente, que por sua vez permite aos usuários do produto fazer seu trabalho de maneira que satisfaça as necessidades de seu negócio” (Robertson, Robertson, 2006, p. 24).

As estatísticas mostram que o investimento em requisitos tem reflexo direto na qualidade dos softwares, sendo que requisitos bem especificados e bem entendidos constituem um dos fatores que mais contribuem para o sucesso dos projetos.

Os processos modernos de desenvolvimento de software buscam trabalhar de forma iterativa e incremental. Por exemplo, a Engenharia de Software possui um processo chamado RUP que atribui tarefas e responsabilidades no desenvolvimento de software baseada em disciplinas através do tempo que é representado por fases, iterações e marcos. Este é um processo de desenvolvimento de software baseado em Casos de Uso, utiliza a UML e portanto usa a metodologia Orientada a Objeto como base. A segunda disciplina deste processo é a de Requisitos, que recebe nas iterações iniciais uma maior atribuição de tempo. Esta disciplina visa além do consenso entre os envolvidos sobre o que o sistema deverá fazer, uma melhor compreensão dos requisitos do sistema e uma base para o planejamento técnico e dimensionamento do mesmo.

2.1 Histórias de Usuário

Para chegar aos requisitos funcionais, as técnicas de desenvolvimento ágil de sistemas como a Xtreme Programing, prevêm reuniões entre especialistas no negócio e desenvolvedores. A intenção não é definir detalhadamente os requisitos e sim descrever histórias de usuário, que representa um compromisso para uma discussão futura sobre os requisitos do software (COCKBURN, 2001, p. 180).

As metodologias ágeis pregam a colaboração com o cliente ao invés de uma negociação de contratos que são representados por especificações detalhadas (DEDIANA, 2008).

O formato clássico para se escrever Histórias de Usuário (COHN, 2004) é apresentado abaixo:

Como um << tipo de usuário >>

Eu quero << uma história >>

Para que << alguma razão >>

Desta forma é definido o usuário envolvido (tipo de usuário), a história (o que deverá ser desenvolvido) e o motivo e importância para o desenvolvimento respectivamente.

Esta técnica procura melhorar a comunicação dos desenvolvedores com os usuários, promovendo um trabalho colaborativo.

2.1.1 Características de uma boa História de Usuário

Bill Wake sugeriu o acrônimo INVEST (COHN, 2004):

Independente, Negociável, Valiosa para o cliente ou usuário, Estimável, Pequena e Testável.

As dependências devem ser evitadas. Suponha 3 histórias semelhantes:

pagamento com Visa, pagamento com Mastercard e pagamento com American Express.

Para evitar estimativas diferentes para este casos, pois os desenvolvedores podem alegar maior facilidade após executado para um dos casos, existem duas maneira para evitar esta dependência, combinando as histórias em uma maior mas independente ou uma maneira diferente para separar as histórias, por exemplo:

- o cliente pode efetuar o pagamento com um dos tipos de cartão
- o cliente pode pagar com dois tipos de cartões adicionais

As histórias devem ser negociáveis. Elas não representam contratos e requisitos que devem ser obrigatoriamente implementados. Os escopos devem ser variáveis, de forma que os desenvolvedores cheguem a um acordo com o cliente sobre o que deverá ser implementado.

As histórias devem ter valor para os clientes e usuário, sendo que histórias com valor somente para os desenvolvedores devem ser evitadas.

É importante para os desenvolvedores poder estimar o tamanho e esforço de desenvolvimento da história. Existem três razões em que a história não é estimável: os desenvolvedores não tem conhecimento do domínio, os desenvolvedores não tem conhecimento técnico e a história é muito grande.

No primeiro caso se os desenvolvedores não entenderam a história por não ter conhecimento do assunto, eles devem conversar com o cliente que escreveu a história. No segundo caso os desenvolvedores não tem como estimar por falta de conhecimento técnico em alguma ferramenta. Neste caso a solução é direcionar um ou mais desenvolvedores para o que o Extreme Programming chama de Spike, que é um breve experimento para aprender sobre uma determinada área.

Histórias podem ser grandes demais, muito pequenas ou de um tamanho apropriado. Histórias grandes demais e pequenas demais são difíceis para serem planejadas. As histórias muito grandes, chamadas épicas, devem ser divididas. As muito pequenas são aquelas que os desenvolvedores não querem escrever ou estimar, pois executá-las representaria um esforço menor. Estas histórias muito pequenas devem ser agrupadas.

As histórias devem ser escritas de forma a facilitar os testes. Histórias não testáveis são normalmente requisitos não funcionais, que são requisitos do software e não da funcionalidade, como por exemplo “o usuário deve achar o software fácil de usar”.

2.1.2 Os 3 C's das Histórias de Usuário

De acordo com Ron Jeffries no artigo “Exential XP: Card, Conversation, Confirmation” (JEFFRIES, 2001), histórias de usuário tem três aspectos críticos os quais são chamados de Cartão, Conversa e Confirmação. Histórias são escritas em cartões e não possuem todas as informações necessárias para se tornar um requisito. No entanto, tem informação suficiente para identificar o requisito e lembrar a todos sobre a história. Os requisitos são comunicados pelos clientes aos desenvolvedores através da conversa. A

conversa acontece o tempo todo, em particular nas reuniões de planejamento e de iteração de planejamento, quando as histórias receberão prazo para implantação.

As histórias implementadas precisam ser confirmadas através de teste de aceitação. Estes testes visam confirmar as funcionalidades requisitadas pelo cliente e são a ele apresentadas para que ele confirme o correto funcionamento.

2.2 Desenvolvimento de Casos de Uso

2.2.1 Conceitos

Casos de Uso

“Um caso de Uso captura um contrato entre stakeholders de um sistema sobre seu comportamento. O Caso de Uso descreve o comportamento do sistema sob diversas condições conforme o sistema responde a uma requisição de um dos stakeholders, chamado de *ator primário*.” (COCKBURN, 2001, p21)

Casos de Uso podem ser apresentados em vários formatos, como textual, fluxogramas, diagramas e outros.

Objetivos

Objetivo é uma propriedade desejada do sistema (ROBSON, ELOFSON, 2004). A maioria dos objetivos de um sistema tem um formato comum, como “o sistema deve fazer X”, onde X é uma certa função que o sistema deve prover (ROBSON, ELOFSON, 2004).

Propriedade de domínio

Propriedade de domínio segundo (ROBSON, ELOFSON, 2004) é definida como sendo independentes de qualquer sistema de software e existindo naturalmente no ambiente

Requisito

É um tipo especial de objetivo que restringe o comportamento do software.

Para ser um requisito, um objetivo deve satisfazer as seguintes três propriedades:

(i) é descrito inteiramente em termos de valores monitorados pelo software, (ii) restringe apenas valores que são controlados pelo software, (iii) os valores controlados não são definidos em termos de valores controlados futuros (ROBSON, ELOFSON, 2004, p1).

Especificação

Tipo especial de requisito - diz respeito somente a propriedades do sistema

2.2.2 Definindo e estruturando objetivos

As necessidades dos stakeholders representam a base para a definição dos objetivos. A partir do detalhamento dos objetivos, o processo de refinamento pode originar os requisitos do sistema. A partir de 25 objetivos a estruturação se torna importante, e com menos pode-se apresentar somente uma lista de objetivos. Na maioria dos sistemas temos uma hierarquia estruturada de objetivos (ROBSON, ELOFSON, 2004).

Os analistas partem dos primeiros objetivos obtidos em reuniões, observações, documentos e sistemas existentes para construir esta hierarquia, derivam novos objetivos utilizando perguntas como? para obter objetivos de mais baixo nível e porquê? para obter objetivos de mais alto nível. Como falamos anteriormente refinando objetivos chegamos aos requisitos do sistema. Para diferenciar objetivo, requisito e especificação, devemos observar o comportamento, ou seja as ações presentes. Os Comportamentos podem ser ambientais, requeridos ou implementáveis. Os Ambientais dizem respeito a propriedades do domínio ou comportamento do ambiente, os Requeridos referem-se a propriedades tanto do ambiente quanto do sistema e os Implementáveis são Comportamentos executados pelo sistema. Uma especificação descreve a forma como o sistema produz os seus comportamentos.

A figura abaixo, extraída do artigo “Use Stories and Use Cases” de Scott Sehlhorst, apresenta diagrama que mostra a realização de objetivos usando Casos de Uso – casos de uso a partir de objetivos, requisitos funcionais a partir dos casos de uso e as especificações de implementação a partir dos requisitos funcionais.

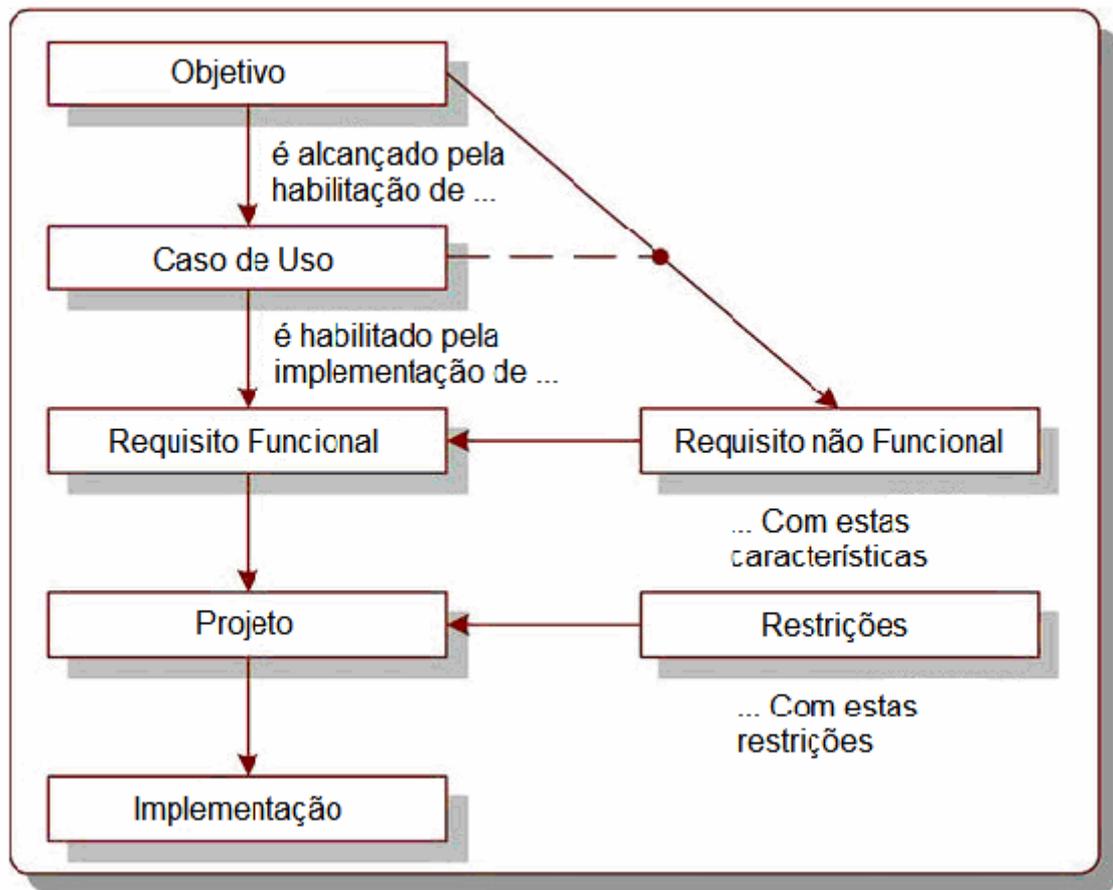


Figura 1 – Casos de Uso permitindo a realização de objetivos
 Traduzido do artigo “Use Stories and Use Cases” de Scott Sehlhorst disponível em
<http://tynerblain.com/blog/2009/02/02/user-stories-and-use-cases/>

2.2.3 Especificando com UML a partir de objetivos

Adicionar objetivos ao método UML acarreta benefícios como abstração, direção, rastreabilidade e análise. (ROBSON, ELOFSON, 2004) definiram um método para obter especificações com UML a partir de objetivos, que consiste das atividades abaixo.

Elicitar o contexto do sistema

Obter informações através de entrevistas, documentos, observação, etc, identificando o contexto do sistema

Definir objetivos do sistema

Utilização do contexto para definir objetivos do sistema

Derivar requisitos

Refinando os objetivos chega-se aos requisitos

Derivar casos de uso

Refinando os requisitos chega-se aos Casos de Uso - Organizacionais, de sistema e de baixo nível

Derivar modelos UML

Outros diagramas UML, como por exemplo "diagrama de classes" são derivados dos requisitos ou Casos de Uso

2.2.4 Refinando objetivos

Patterns de refinamento

A hierarquia de objetivos é criada através do refinamento dos objetivos. Para refinar objetivos temos que acrescentar detalhes a estes objetivos.

Patterns de refinamento básicos podem ser de disjunção ou conjunção. A Disjunção especifica alternativas para satisfazer um objetivo, enquanto conjunção detalha a descrição de um objetivo. Quanto mais detalhes específicos forem fornecidos maior será a hierarquia de objetivos.

Quanto ao padrão de refinamento podem ser marco ou baseado em casos.

Marco - "Decompõem o que se obtém num conjunto de etapas intermediárias. A soma destas etapas satisfazem o objetivo global."

Baseado em casos - "Detalha o resultado num conjunto de casos, que se somam para satisfazer o objetivo total. "

Podemos refinar um objetivo efetuando a pergunta "Como pode este objetivo ser satisfeito?", o que expande a hierarquia de forma descendente, especializando o objetivo. Já com a pergunta "Por que é que isto é um objetivo do sistema?" expande a hierarquia na direção oposta levando a objetivos mais abstratos.

2.2.5 Priorizando objetivos

Alistair (COCKBURN, Escrevendo Casos de Uso Eficazes, 2001) propõe que se poupe energia ao escrever um caso de uso dividindo o processo em quatro estágios de precisão, de acordo com a energia necessária para cada estágio e o valor da pausa após cada estágio:

1. Atores e objetivos – liste, revise a exatidão e a completude da lista e priorize.

2. Caso de uso breve ou cenário de uso principal – esboce cenário de sucesso principal, revise no formato rascunho quanto ao real atendimento aos interesses dos stakeholders. Isto representa o segundo nível de precisão nos requisitos funcionais.
3. Condições de falha – complete o cenário de sucesso principal e levante todas as falhas que podem ocorrer. Liste completamente as falhas antes de trabalhar em como o sistema deve tratá-las.
4. Escreva como o sistema deve responder a cada falha.

2.2.6 Escopo

“Escopo é a palavra que utilizamos para a extensão do que desenvolvemos em oposição ao trabalho de desenvolvimento de outra pessoa ou algo já desenvolvido.” (COCKBURN, 2001, p49).

Temos o Escopo funcional e o Escopo de desenvolvimento. São apresentadas 3 ferramentas para auxiliar na delimitação do escopo funcional: a lista dentro/fora, a lista ator-objetivo e os resumos de Caso de Uso.

A lista dentro/fora consiste em uma tabela com 3 colunas – tópico, dentro e fora. Em reuniões sobre requisitos, no caso de dúvidas um tópico é acrescentado na tabela e submetido à opinião de todos. Se não existe consenso, um apelo ao comitê de direção do projeto se fará necessário (COCKBURN, 2001). Na lista ator-objetivo os objetivos dos atores são listados e são atribuídas as prioridades para estes objetivos.

“O Resumo de Caso de Uso é uma descrição de duas a seis sentenças do comportamento do Caso de Uso, mencionando apenas as atividades e falhas mais significantes” (COCKBURN, 2001, p51).

Ao contrário do escopo funcional, o escopo de desenvolvimento não é adequadamente definido através da lista ator-objetivo e dos casos de uso. Se não especificado, um mesmo conjunto de casos de uso podem nos levar a sistemas de diferentes abrangências. Ou seja, quem está lendo um caso de uso poderá considerar o sistema com uma abrangência maior ou menor do que era a intenção do autor. Para facilitar o entendimento Alistair (COCKBURN,

2001) rotulou os Casos de Uso com o escopo de desenvolvimento, destacando esta informação através de ícones. Ele atribuiu um ícone de casa para negócio, uma caixa para sistema e um parafuso para sub-sistema ou componente.

2.2.7 Níveis de Casos de Uso

Alistair (COCKBURN, 2001) também nomeou três níveis de objetivos. Nível de resumo (ícone de pipa ou nuvem se muito alto – cor branco), nível de objetivos do usuário (ícone nível do mar – cor azul), e nível de subfunções (ícone peixe / cor índigo – ícone marisco / cor preto). Destes níveis o nível do usuário é o que deve ser procurado, pois é ele que importa para o ator primário, ou seja, ele representa o objetivo do usuário. Para encontrar este nível temos que focar no que o ator primário deseja do sistema. Se não for exatamente o que o ator primário quer, temos que fazer as perguntas para refinamento (Porquê? Como?) para subir ou descer respectivamente o nível até chegar ao nível do usuário.

2.2.8 Formatos de Casos de Uso

Vários fatores podem influenciar na escolha do formato para a escrita de Casos de Uso. Existe a preferência do autor por determinado formato, porém dependendo do momento e das pessoas envolvidas, a escolha de outro estilo poderá ser mais adequada. A quem se destina o caso de uso que está sendo escrito? É para a leitura de um stakeholder? Será destinado a um implementador? Qual é o nível de conhecimento domínio? O assunto é complexo? Além destes, vários outros fatores podem influenciar na escolha de um formato. A seguir apresentamos formatos disponíveis para a escrita de um caso de uso.

Formatos para escolher (COCKBURN, 2001, p122)

Inteiramente Completo, Casual, Tabela de uma Coluna, Tabela de duas Colunas, Estilo RUP, Estilo de Expressões Se, Estilo Occam, Estilo de Diagrama, O Diagrama de Caso de Uso da UML.

O Inteiramente Completo possui um coluna de texto (não uma tabela), os passos são numerados, não possuem a sentença “Se” e utilizam uma convenção numérica na seção de extensões que envolve combinações de dígitos e letras (por exemplo, 2a, 2a1, 2a2, e assim por diante). O formato Casual é uma descrição de texto casual onde é descrito inclusive as

extensões. A tabela de uma coluna inclui os passos do cenário em tabela, apenas como forma de apresentação ou visualização. A tabela de Duas Colunas posiciona as ações do ator primário na coluna da esquerda e as ações do sistema na coluna direita. É usada mais frequentemente na preparação para o projeto de interface com o usuário. O Estilo RUP utiliza um template semelhante ao utilizado no formato inteiramente completo. Nele as extensões são chamadas de “fluxos alternativos” e a numeração dos passos é opcional. O Estilo de Expressões “Se” utiliza expressões se no texto e é normalmente preferido pelos programadores em contrapartida a escrever extensões. O Estilo Occam é uma escrita formal com linguagem Occam inventada por Tony Hoare. Esta linguagem permite que sejam escritas seqüências alternativas, paralelas e opcionais, mas falta saber como são tratadas as exceções (COCKBURN, 2001). No Estilo Diagrama pode-se utilizar vários diagramas como de seqüência, de colaboração, de atividade e redes Petri. Por último o Estilo Diagrama de Caso de Uso da UML que utiliza Elipses, setas e bonecos de palito.

2.3 Cenários

Um conjunto de eventos constituem cenários em casos de uso. Um caso de uso pode conter várias seqüências, ou seja, vários cenários diferentes. Cada cenário se inicia com uma determinada condição ou acionamento e pode terminar atingindo ou não o objetivo.

O Cenário é composto de ações de vários atores para atingir o objetivo. Os passos de ação podem ocorrer em paralelo, em qualquer ordem, podem ser repetidos e podem ser opcionais (COCKBURN, 2001, p95). Os passos do cenário podem descrever uma interação entre atores ou uma ação para proteger algum interesse de um stakeholder. Como exemplo de passo de ação podemos citar:

“O sistema verifica a validade do CPF”

“O cliente entra com o pedido”

A numeração ou não dos passos de ação é uma opção de quem escreve os casos de uso. Manter a numeração pode ser uma tarefa trabalhosa, porém facilitam a referência às extensões.

2.3.1 Cenário de sucesso principal

Cenário no qual o objetivo do ator primário é alcançado e todos os interesses dos Stakeholders são satisfeitos (COCKBURN, 2001). De um modo geral, o cenário de sucesso principal não

possui nenhuma condição ou desvio, o que não seria errado ou ilegal, mas que é preferível deixar todo o tratamento condicional para as Extensões (LARMAN, 2005).

O cenário de sucesso principal não é o único cenário que alcança sucesso. Outros cenários de sucesso são descritos nas extensões, assim como o tratamento para falhas.

2.3.2 Extensões

São cenários escritos para cada situação de desvio que ocorram no cenário de sucesso principal. Extensões representam normalmente a maioria do texto e abrangem todos os casos que se ramificam do cenário de sucesso principal, tanto de sucesso como de fracasso. A combinação entre caminho correto e cenários de extensão devem satisfazer a quase todos os interesses dos interessados (LARMAN, 2005).

Para escrever as extensões existem três opções: escrever cada um dos cenários separadamente, escrevendo expressões *se* e escrever o cenário de sucesso principal com uma seqüência simples do acionamento até a conclusão e um cenário de extensão para cada ponto de desvio. Esta última se apresenta como a melhor opção (COCKBURN, 2001, p105).

3 ANÁLISE DAS TÉCNICAS APRESENTADAS

3.1 Análise comparativa das técnicas apresentadas

“História de Usuário é uma descrição de objetivos centrados no usuário que uma ou mais pessoas irá alcançar usando seu produto” (SEHLHORST , 2009).

Nos processos ágeis as Histórias de Usuário representam o meio pelo qual chegamos aos requisitos a partir dos objetivos dos Stakeholders. Na abordagem com Requisitos estruturados, os casos de uso são a ligação entre os objetivos dos Stakeholders e os requisitos funcionais. Portanto, as duas técnicas cumprem funções semelhantes. De acordo com Scott Sehlhorst temos Histórias de Usuário, Cenários de Caso de Uso, Casos de Uso formal , Casos de Uso informal e Casos de Uso breves. Quanto mais formal, maior o nível de detalhes capturado e maior o esforço de documentação. Isto é ilustrado na figura 2 extraída do artigo “User Stories and Use Cases” de Scott Sehlhorst. As tecnologias ágeis dão maior importância à comunicação em detrimento de uma documentação mais abrangente. A quantidade requerida de reuniões e conversas entre os especialistas no negócio e os desenvolvedores para um entendimento satisfatório das histórias de usuário irá depender do domínio que a equipe de desenvolvimento já possui do assunto (SEHLHORST , 2009). A determinação da suficiência das reuniões para a completude das histórias de usuário representa uma dificuldade para as equipes. Através de pesquisas feitas por Alistair por vários anos (SEHLHORST , 2009), ele constatou que as equipes se deparam com problemas semelhantes: entendimento incompleto do contexto do objetivo almejado pelo usuário, a equipe não consegue determinar com antecedência o escopo do projeto e os comportamentos alternativos dos usuários não são identificados antes do compromisso de entrega. Para solucionar estes problemas sugere a utilização de Casos de Uso. A argumentação feita por Sehlhorst é que apesar dos argumentos a favor da utilização de Casos de Uso para resolver estes problemas estarem corretos, você será muitas vezes levado a criar histórias de usuário. Ao trabalhar com histórias de usuário o desenvolvedor deverá ser capaz de resolver os problemas apontados por Cockburn. Isto pode ser feito com verificações e um pequeno acréscimo na quantidade de documentação.

“O Caso de Uso descreve o comportamento do sistema sob diversas condições conforme o sistema responde a uma requisição de um dos stakeholders” (COCKBURN, 2001, p21).

Casos de uso capturam maior nível de detalhes que histórias de usuário, sendo um processo com uma documentação mais completa. Por outro lado, histórias de usuário é uma técnica mais rápida e mais barata.

A opção por uma ou outra pode ser definida em função da complexidade da aplicação a ser desenvolvida. A avaliação do nível de complexidade do que se vai desenvolver não pode se limitar à análise focada unicamente no que os usuários querem fazer. Esta avaliação deve levar em consideração a capacitação técnica da equipe de desenvolvimento, bem como o conhecimento já adquirido por esta equipe sobre o domínio do negócio (SEHLHORST, 2009). Podemos ter também as duas técnicas utilizadas em conjunto.

O custo de um maior esforço de documentação tem um benefício - mais detalhe.

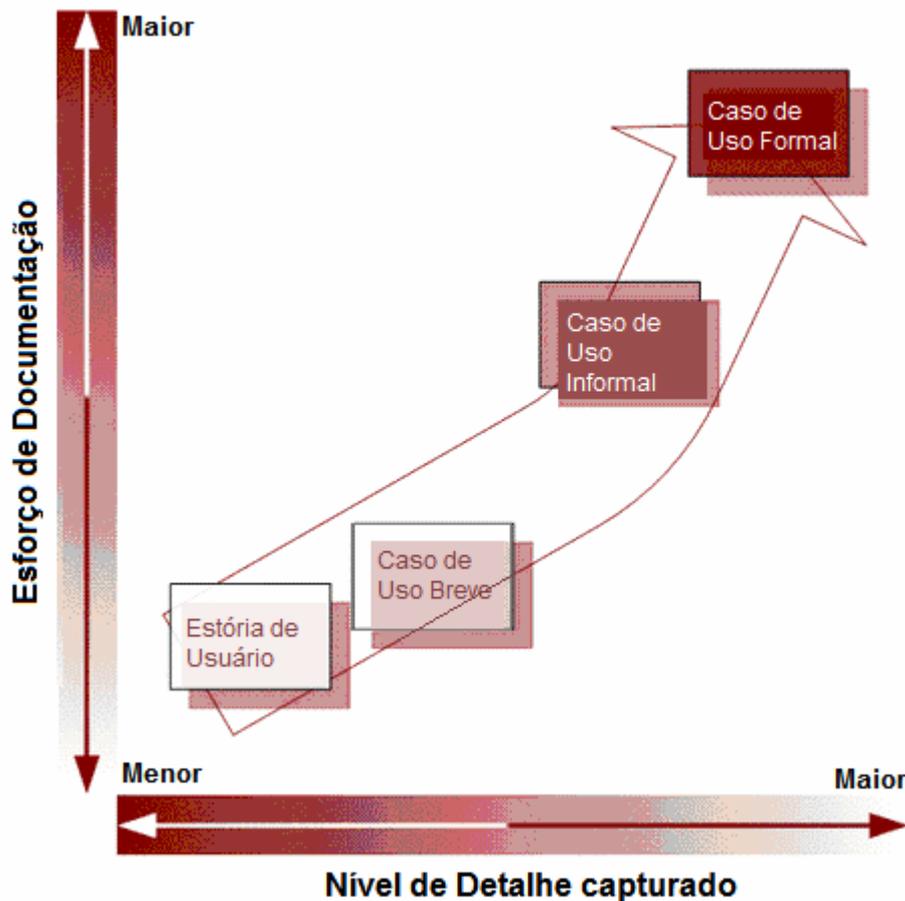


Figura 2 – Formalidade, esforço de documentação e detalhes capturados
Traduzido do artigo “Use Stories and Use Cases” de Scott Sehlhorst disponível em <http://tynerblain.com/blog/2009/02/02/user-stories-and-use-cases/>

4 ESTUDO DE CASO

Este estudo de Caso aborda um projeto de desenvolvimento de uma funcionalidade típica para cartões de crédito. A nova funcionalidade deverá representar uma customização a ser implementada no Sistema de Gerenciamento de Crédito. A intenção aqui não é apresentar um projeto completo e detalhado, mas mostrar a aplicação das técnicas estudadas. Os requisitos funcionais também serão apresentados nos moldes hoje utilizados nesta área de negócios, para efeito de evidenciar as diferenças e opções na apresentação e documentação dos requisitos levantados.

4.1 Descrição do Problema

O Estudo de Caso é de um projeto que tem por finalidade atribuir um percentual adicional de OverLimit temporário a ser aplicado sobre o Limite de Crédito de contas de clientes portadores de cartão de crédito selecionados em uma campanha promocional . O Sistema em questão deverá avaliar a situação momentânea da conta para concessão ou não deste crédito.

Abaixo temos um documento com os requisitos funcionais para o projeto, levantados e documentados em um formato de uso comum no setor. Ele normalmente é derivado de um levantamento de requisitos de negócio, elaborado anteriormente a partir de reuniões e entrevistas com os demandantes.

Documento de Especificação Funcional

1. Cliente Solicitante

Nome

Departamento

Telefone

2. Descrição Geral

A Administradora de Cartões de Crédito do Banco XYZ solicita que seja possível a atribuição de um percentual extra de OverLimit temporário para portadores de cartões

que serão alvo de campanha promocional para extensão no limite de crédito para um período pré determinado. Este percentual representará um incremento ao que atualmente é atribuído e mesmo que o cliente já seja beneficiado com percentual de overlimit diferenciado em função de comportamento e relacionamento. O banco irá selecionar os clientes participantes. Findado o prazo da campanha, dependendo de critérios de avaliação quanto à utilização desta linha de crédito, o sistema poderá renovar a aplicação deste percentual. O cliente portador de cartão selecionado para a campanha deverá receber carta informativa. A funcionalidade poderá ser novamente parametrizada em futuras campanhas.

3. Escopo da Solicitação

Criar parâmetros para campanha, receber relação de clientes, processar critérios de avaliação da conta do cliente, atribuir novo limite temporário para contas confirmadas pela avaliação, gerar relatório com contas com situação impeditiva, controlar prazo de expiração e enviar carta de comunicação ao cliente.

4. Objetivo

Incentivar maior utilização do cartão por portadores que representem pequeno risco de inadimplência, concedendo um percentual além da política padrão do Banco, porém por um período determinado para a campanha.

5. Restrições

A funcionalidade será disponibilizada para os produtos ou tipos de cartões de forma parametrizável a critério do banco.

6. Descrição Situação Atual

O percentual de Overlimit para efeito de autorizações de movimentações financeiras no cartão de crédito é atribuído através de parâmetro por tipo de cartão aplicado sobre o limite de crédito da conta. O sistema também possui uma tabela de parâmetros de percentuais especiais que são aplicados incrementando o limite da conta de acordo com o perfil de utilização/comportamento do cliente.

7. Situação Desejada

Atribuir percentual extra de overlimit através de parâmetro por tipo de cartão para campanha promocional. As contas selecionadas deverão passar por críticas em relação a bloqueios – criar uma matriz de bloqueios que será parametrizada como impeditivos para atribuição do percentual de overlimit.

Controlar o período de validade da campanha para retorno à situação anterior ou renovar a aplicação do percentual extra.

Após o fim de uma campanha, a qualquer momento poderá ser parametrizada uma nova a critério do Banco.

8. Requisitos Funcionais

RF 1 – Criar novo parâmetro no nível de produto/tipo de cartão que indique que a funcionalidade está ativa.

RF 2 – Criar nova tela para parâmetros também no nível de produto/tipo para atendimento à campanha que deverá conter:

Data início e data fim de validade

Valor do incremento ao percentual de Overlimit existente

Matriz de bloqueio de A a Z, que receberão valores “S” (impeditivo para a campanha) ou “N” (não impeditivo para a campanha)

Indicador se renovável

RF 3 – As contas selecionadas e enviadas pelo Banco devem ser avaliadas quanto à existência de algum bloqueio impeditivo para a campanha (um bloqueio impeditivo pode ter sido atribuído à conta em um momento posterior à seleção pelo Banco)

RF 4 - As contas da campanha que não possuem impedimento deverão receber um novo campo de “Percentual de Overlimit de campanha” que irá incrementar o valor do limite após a aplicação do Overlimit normal da conta.

Ex.: Limite de crédito – R\$ 4.000,00

Percentual de Overlimit aplicado à conta atualmente = 13%

Limite atual para efeito de autorizações = 4.000,00 + 13% = R\$ 4.520,00

Incremento da campanha

Valor de incremento parametrizado = 10 %

Limite na campanha para autorizações = 4.520,00 + 10% = R\$ 4.972,00

RF 5 – Atribuir às contas contempladas o período de validade do percentual igual ao período da campanha atual, bem como o indicador se este pode ser renovado.

RF 6 – Expirado a validade do percentual extra o sistema deverá verificar a situação das contas contempladas na campanha. Uma vez o indicador de renovação igual a sim, o sistema deverá conceder a prorrogação da aplicação do percentual extra, mas somente para as contas adimplentes.

Cabe aqui ressaltar que o Sistema de Gerenciamento de Crédito é um pacote de sistemas implementado em mainframe, de grande porte e bastante complexo. É um sistema que atende a múltiplas Organizações (processa cartões de crédito de vários bancos e de diversas bandeiras – Visa, Mastercard, American Express e cartões Private Label, simultaneamente). Por exemplo, uma demanda que interfere na área monetária e contábil normalmente não pode ser desenvolvida em módulo apartado e simplesmente se integrando ao módulo monetário do sistema. Estes tipos de demanda representam customizações no módulo existente e um maior detalhamento dos requisitos funcionais se torna necessário. Normalmente os requisitos chegam à equipe de desenvolvimento nos níveis de usuário e/ou detalhado, mas sempre no formato textual apresentado no exemplo acima.

As metodologias voltadas para desenvolvimento rápido prevêm reuniões com o intuito de descrever histórias de usuário escritas por ele na sua própria linguagem. A intenção não é detalhar os requisitos, passo este que deverá ocorrer posteriormente. Levando-se em conta o que foi relatado no parágrafo anterior, vejo que a utilização de histórias de usuário não se torna muito adequada à característica do sistema em questão, uma vez que requer um nível de detalhamento e documentação maior. A utilização desta técnica implicaria em mudanças na metodologia de desenvolvimento das Empresas processadoras de cartão, pois dependeriam de fases de iterações e confirmações que hoje não são praticadas. A proposição de mudanças neste sentido fogem do intuito deste trabalho. Porém, para exemplificar a técnica relacionamos abaixo algumas histórias de usuário que poderiam ser escritas para o projeto.

h1 – “Como membro da Diretoria do Banco XYZ desejo incentivar os portadores de cartões de crédito a intensificar o seu uso, visando aumento da receita da XYZ Cartões”

h2 – “A Diretoria do Banco XYZ deseja definir critérios de seleção de contas alvo de campanha de incentivo”

h3 - “Em nome da Administradora XYZ Cartões quero selecionar clientes bons pagadores que hoje utilizam mensalmente mais da metade do seu limite de crédito para incentivá-los a usar ainda mais o cartão”

h4 – “O Banco XYZ quer definir parâmetros de controle para a campanha”

h5 – “O Banco XYZ deseja ampliar o limite do cliente selecionado através de um percentual parametrizado para a campanha”

h6 – “O portador do cartão recebe carta com informações da campanha”

As histórias não contemplam detalhes como quais são os critérios de seleção, o que é considerado bom pagador para a campanha? Quem paga todo mês a fatura integral ou quem

utiliza o crédito rotativo? Quais serão os parâmetros de controle da campanha como período de validade, percentual de OverLimit extra a ser concedido, critérios de exceção? Também não leva em consideração como o sistema deve proceder para recalcular o novo limite de crédito para efeito de autorizações, o conteúdo da carta que o cliente irá receber e etc. As histórias servem de base para que em novas reuniões planejadas, os detalhes dos requisitos possam ser captados pelos analistas.

O trabalho prossegue com a utilização de casos de uso. Casos de uso são hoje muito utilizados nos métodos de análise orientados a objetos. Para o ambiente do projeto apresentado falamos de um sistema de grande porte implementado de forma Estruturada utilizando linguagem COBOL, em ambiente CICS e arquivos VSAM. Os projetos de desenvolvimento nesta área representam customizações e agregações ao sistema legado. O trabalho desta monografia busca apresentar a utilização de Casos de Uso como ferramenta para especificação de requisitos destes projetos como vantajosa e viável, podendo ser complementar ou substitutiva para o formato hoje utilizado.

O processo para elicitar o contexto do projeto é o mesmo utilizado atualmente, ou seja, através de entrevistas, reuniões, documentos, etc.

Vamos definir os objetivos a partir do contexto, derivar os requisitos a partir do refinamento dos objetivos e derivar os casos de uso a partir do refinamento dos requisitos. Podemos evidenciar Casos de Uso no nível resumo, de usuário e de subfunções.

Lista de objetivos para o Estudo de Caso

- A Administradora de Cartões do Banco XYZ deseja atribuir percentual extra de OverLimit para determinados clientes
- A Administradora deve estipular o percentual a ser aplicado
- A Administradora deve estipular o período de validade da campanha
- O Sistema deverá aplicar o percentual de Overlimit extra na conta de clientes
- A Administradora de Cartões deve definir os clientes que participarão da campanha
- O Cliente deverá receber carta informativa
- A Administradora deseja fazer outras campanhas futuras
- O Sistema deverá avaliar a conta para renovação do percentual extra no final do período da campanha

Perguntas porquê e como? são feitas para encontrar os objetivos de nível superior e inferior respectivamente. Por exemplo, para o objetivo “A Administradora de Cartões do

Banco XYZ deseja atribuir percentual extra de OverLimit para determinados clientes” chegamos aos objetivos abaixo:

Por quê? – “A Administradora quer estimular um maior uso do cartão pelos portadores”

Como? – “deve selecionar clientes alvo ”

No quadro 1 abaixo temos a lista dos objetivos identificados separada por ator e sistema, com características abstrato, no nível de usuário e sem exceções.

QUADRO 1 Lista de Objetivos por Ator e Sistema

Objetivos do Ator	Objetivos do Sistema
Administradora quer estimular um maior uso do cartão pelos portadores	
Administradora quer atribuir percentual extra de OverLimit	
Administradora deve selecionar clientes alvo	
Administradora deve estipular o percentual de Overlimit extra a ser aplicado	
Administradora deve definir o período de validade da campanha	
	O sistema deve receber a relação de clientes para a campanha
	O sistema deve avaliar critérios restritivos para aplicar o percentual à conta
	O sistema deve aplicar o percentual de Overlimit extra na conta de clientes
	O sistema deve enviar carta informativa ao cliente
	O Sistema deverá avaliar as contas para renovação do percentual extra no final do período da campanha

Obtendo os Requisitos para os objetivos levantados.

1. Administradora quer estimular um maior uso do cartão pelos portadores

Como?

- a. Administradora deseja atribuir percentual extra de OverLimit

Como?

Administradora seleciona clientes adimplentes e com comprometimento atual de mais de 50% do limite de crédito

Como?

- Administradora pesquisa a base de dados do sistema e seleciona clientes
- Administradora envia para o sistema um arquivo contendo clientes selecionados para campanha

- b. Administradora parametriza percentual de OverLimit extra a ser aplicado

Como?

- Administradora inclui na tela de parâmetros da campanha o percentual de Overlimit extra a ser aplicado

- c. Administradora parametriza período de validade para a campanha

Como?

- Administradora inclui na tela de parâmetros da campanha a data de início e a data de fim da campanha

- d. Administradora parametriza códigos de bloqueios que impedem aplicação de percentual sobre o limite de crédito do cliente

Como?

- Administradora inclui na tela de parâmetros os códigos de bloqueios de contas impeditivos para a campanha

2. O sistema deve receber a relação de clientes para a campanha

Como?

- Sistema recebe arquivo de clientes para processamento

3. O sistema deve avaliar critérios restritivos para aplicar o percentual à conta

Como?

- Sistema lê a conta na base de dados e verifica se conta possui código de bloqueio e se este está parametrizado como restritivo para aplicar o percentual extra de OverLimit à conta.

Bloqueio impeditivo

Relaciona motivo e envia à Administradora

Ou

Sem bloqueio impeditivo

Aplica percentual extra de OverLimit na conta

4. O sistema deve aplicar o percentual extra de Overlimit na conta de clientes

Como?

a. Sistema acrescenta percentual extra de OverLimit ao limite de crédito existente na conta de clientes

Como?

- Sistema lê base de parâmetros da campanha
- Sistema lê na base de dados a conta selecionada, acrescenta percentual parametrizado ao seu limite de crédito e atualiza base

5. O sistema deve enviar carta informativa ao cliente

Como?

- Sistema envia carta informativa ao cliente que teve limite de crédito aumentado na base de dados

6. O Sistema deverá avaliar as contas para renovação do percentual extra no final do período da campanha

Como?

- Sistema acessa a base de contas e renova percentual extra no final da campanha para contas adimplentes

No quadro 2 temos uma hierarquia de requisitos com característica concreto, nível do usuário e com exceções

QUADRO 2 Hierarquia de Requisitos Concreto, nível do Usuário e com exceções

Requisitos do Ator	Requisitos do Sistema
Administradora pesquisa base de dados do sistema e seleciona clientes	
Administradora envia para o sistema um arquivo contendo clientes selecionados para campanha	
Administradora inclui na tela de parâmetros da campanha o percentual de Overlimit extra a ser aplicado E Administradora inclui na tela de parâmetros da campanha a data de início e a data de fim da campanha E Administradora inclui na tela de parâmetros os códigos de bloqueios de contas impeditivos para a campanha	
	Recebe arquivo de clientes para processamento
	Lê a base de dados e verifica existência de código de bloqueio impeditivo para campanha na conta Bloqueio impeditivo Relaciona motivo e envia à Administradora Ou Sem bloqueio impeditivo Sistema acrescenta percentual extra de OverLimit parametrizado ao limite de crédito existente na conta de

	clientes E envia carta informativa ao cliente
	Sistema lê a base de dados e renova percentual extra no final da campanha para contas adimplentes E Envia carta ao Cliente OU Retira percentual extra de OverLimit

Mapas Mentais também são usados para demonstrar a Hierarquia de Objetivos/Requisitos do sistema conforme figura 3 com o mesmo exemplo de hierarquia anterior que é exemplificado com o uso do CMAP

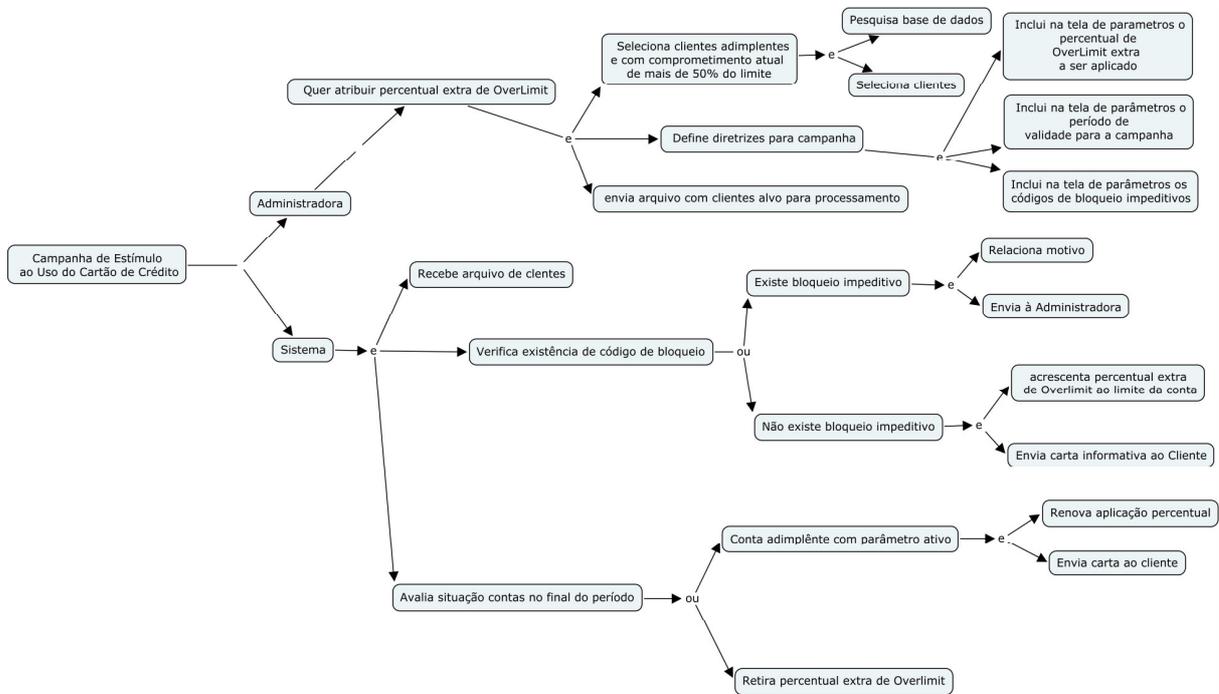


Figura 3 – Uso de mapas mentais para representar hierarquia de objetivos/requisitos criado utilizando software CMAP TOOLS – <http://cmap.ihmc.us/>

Perguntas Por quê? e Como? foram feitas anteriormente elevando ou abaixando os níveis de objetivos. Agora podemos elaborar Casos de Uso a partir da hierarquia de objetivos, mas antes devemos marcar os objetivos com o seu nível correspondente. Desta forma podemos elaborar Casos de Uso no nível desejado. Para esta marcação tomamos o nível do Usuário como base, portanto ele deve ser identificado. Embora subjetivo, a identificação do nível do usuário pode ser feita através da pergunta “É isso que realmente o ator primário quer do sistema agora? (COCKBURN, 2001, p79)

Objetivos, requisitos e especificações marcados por níveis através dos ícones propostos por Alistair (Escrevendo Casos de Uso Eficazes, 2001)

Efetuar Campanha

Administradora quer estimular um maior uso do cartão pelos portadores 

- Administradora quer atribuir percentual extra de OverLimit 
- Pesquisa base e seleciona clientes alvo 
- Define diretrizes para campanha 
- Inclui na tela parâmetros percentual Overlimit extra 
- Inclui na tela parâmetros período de validade 
- Inclui na tela códigos de bloqueios restritivos 
- Envia arquivo de clientes alvo para processamento 
- Sistema recebe arquivo de Clientes 
- Sistema verifica a existência de código de bloqueio 
- Existe bloqueio impeditivo
 - Relaciona motivo 
 - Envia à Administradora 
- Não existe bloqueio impeditivo
 - acrescenta percentual extra de Overlimit ao limite da conta 
 - Envia carta informativa ao cliente 
- Sistema avalia situação de contas no final do período para renovação 
- Conta adimplente e parâmetro renovação ativa
 - Renova percentual extra de Overlimit 
 - Retira percentual extra de Overlimit 

Legenda dos ícones propostos por Alistair (Escrevendo Casos de Uso Eficazes, 2001) para marcar os Níveis de objetivos

-  - Estratégico muito alto
-  - Estratégico
-  - Nível do Usuário
-  - Detalhe
-  - detalhe muito baixo

A figura 4 mostra Casos de Uso marcados no nível do Usuário

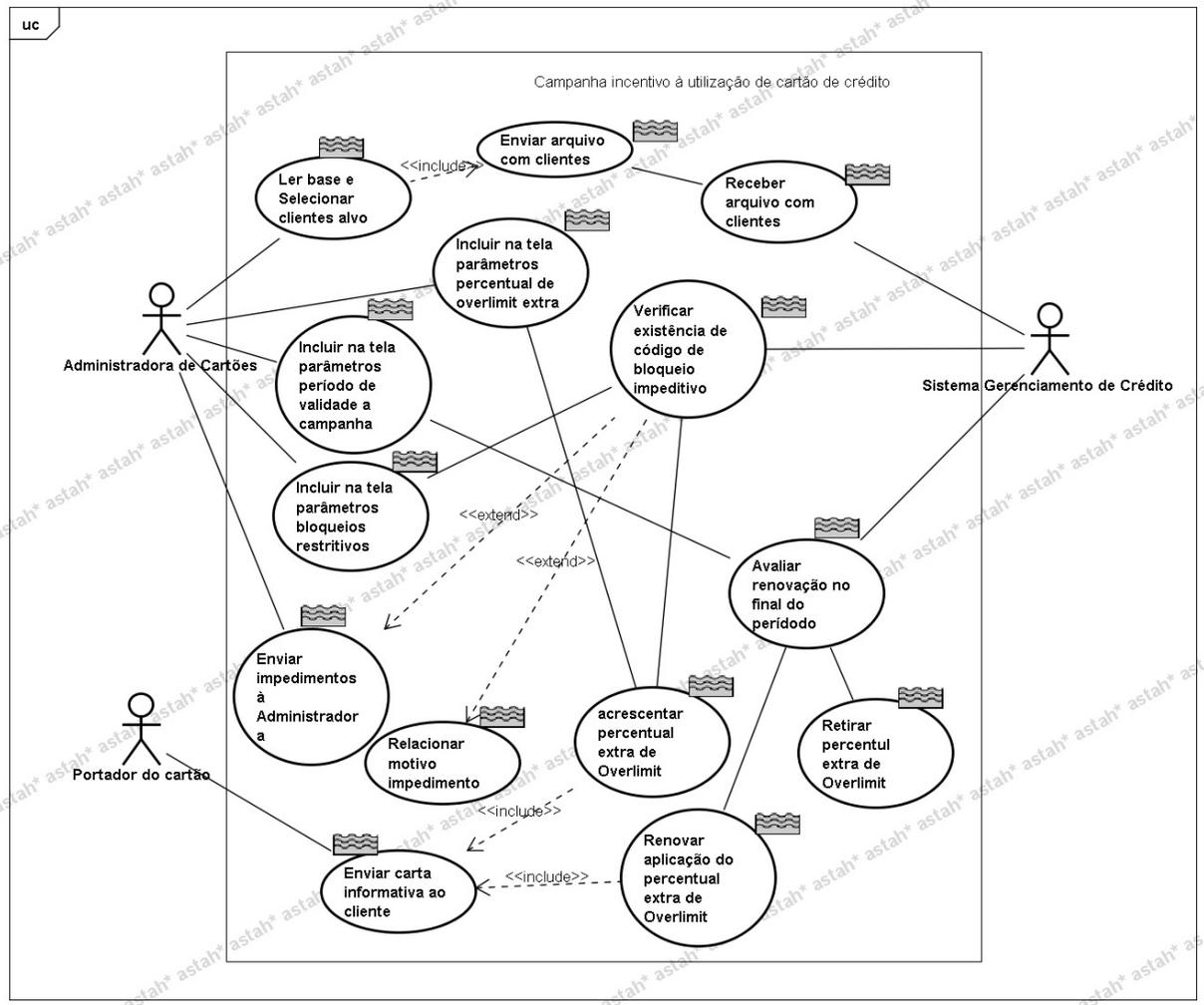


Figura 4 – Casos de Uso marcados no nível do Usuário criado utilizando o software astah community - <http://jude-users.com/en/>

Segue um exemplo de um caso de uso escrito no formato Completo.

Caso de Uso Verificar existência de código de bloqueio impeditivo

Contexto de Uso: Sistema confronta código de bloqueio da conta em relação a matriz de bloqueios impeditivos parametrizados pela Administradora de Cartões de Crédito

Escopo: Módulo de controle da Campanha de Incentivo ao uso de cartões

Nível: Objetivo do Usuário

Ator Primário: Sistema de Gerenciamento de Crédito

Stakeholders e Interesses: Banco XYZ

Pré-condição: Conta selecionada

Garantias mínimas: cliente sem restrições para a linha de crédito

Garantias de sucesso: Contas avaliadas para aumento no limite de crédito

Acionador: Sistema de Gerenciamento de Crédito recebe arquivo de clientes da campanha

Cenário de Sucesso Principal:

1. Sistema acessa conta constante do arquivo de clientes da campanha
2. Sistema acessa matriz de parâmetros de bloqueios impeditivos
3. Sistema verifica a inexistência na conta de bloqueio impeditivo

Extensões:

- 3.a. Sistema identifica bloqueio impeditivo
 - 3.a.1 Sistema relaciona o motivo de impedimento
 - 3.a.2 Sistema envia impedimento à Administradora de Cartões

Lista de Variações tecnológicas & de Dados

Informação relacionada

5 CONCLUSÃO

A minha intenção inicial para elaboração deste trabalho era selecionar um projeto piloto para utilizar as técnicas estudadas durante a fase de Requisitos, o que faria deste um trabalho empírico e possibilitaria comparações com outros projetos semelhantes. Isto, no entanto, não se viabilizou devido a não me encontrar mais na condição de funcionário destas empresas e sim como prestador de serviços. Sendo assim, optei pelo desenvolvimento de um Estudo de Caso para exemplificar a utilização das técnicas.

As especificações de requisitos para agregações ao sistema legado que checam hoje com erros ou incompletas para as equipes de desenvolvimento, não estão relacionadas exatamente com o formato em que são apresentadas, mas devido a não se aplicar técnicas adequadas na captura dos requisitos. É fato também que o formato utilizado, dependendo do tamanho do projeto, dificulta a leitura e muitas vezes detalhes passam a depender bastante da habilidade para descrição de quem elabora o documento, facilitando ou não a sua interpretação. Conforme já comentado anteriormente, o método de desenvolvimento utilizado nesta área divide claramente o processo com etapas em “Cascata”. Neste processo cada fase ou etapa é baseada no produto resultante da anterior, onde temos as fases de Requisitos, Análise, Design, Construção, Teste e Manutenção. O que ocorre nas citadas empresas é a comunicação acontecendo quase sempre somente entre a equipe da fase atual e a equipe da fase anterior mais próxima. Erros ocorridos nas fases precedentes são percebidos tardiamente, ocasionando alto custo para correção.

Vemos que uma mudança de metodologia para um processo moderno, onde iterações com liberações parciais de implementações de requisitos resultam em avaliações, resultados e correções mais rápidas, seria o melhor caminho para o setor proporcionando uma satisfação maior dos demandantes. Mas, reforçando o que já foi dito, a intenção do trabalho não é propor uma metodologia específica e sim confirmar a viabilidade e benefício de usar as ferramentas estudadas mesmo que estas empresas não façam uma mudança significativa nos métodos que utilizam.

A técnica de “Histórias de Usuário”, conforme vimos no item 2.1 deste trabalho, fazem de descrições pouco detalhadas compromissos para reuniões futuras visando um maior esclarecimento das histórias. Neste mesmo item vimos também que as Histórias de Usuário não representam um contrato de desenvolvimento, sendo negociável com o cliente o que será implementado. Estas características implicam em trabalho cooperativo entre a equipe de

desenvolvimento e os demandantes, sendo que não está implementada nas empresas citadas uma metodologia de desenvolvimento que favoreça o seu emprego e que o torne interessante e vantajoso.

As documentações de Requisitos chegam hoje para as equipes de desenvolvimento das Processadoras em um formato textual livre, mais detalhadas e com a aprovação das Administradoras de Cartões. Estas documentações aprovadas representam contratos neste processo de desenvolvimento típico “em Cascata” e são passados para a equipe da próxima fase para cumprimento do mesmo. “Casos de Uso”, trabalhado no item 2.2, constitui uma técnica mais estruturada e que captura mais detalhes a medida que é empregada uma maior formalidade – visto no item 3.1 “Análise comparativa das técnicas apresentadas”. Mesmo sem a existência nestas empresas de uma metodologia moderna voltada para um processo iterativo e incremental, Casos de Uso se apresentam como uma técnica viável e perfeitamente aplicável à realidade atual deste tipo de sistema legado. Conforme apresentamos no mesmo já citado item 3.1, Alistair constatou problemas semelhantes em projetos que requeriam um maior detalhamento e onde se substituiu casos de uso por histórias de usuário: entendimento incompleto do contexto do objetivo almejado pelo usuário, escopo não identificado com antecedência e comportamentos alternativos sendo identificados somente após o prazo acordado com o cliente.

A utilização de “Histórias de Usuário”, considerando que as demandas de aplicações chegam com diversos níveis de complexidade, pode se adequar a determinados projetos, apesar das ressalvas apresentadas anteriormente, mas para tanto uma documentação um pouco mais abrangente se torna necessária.

Consideramos portanto que “Casos de Uso” é uma técnica mais adequada para emprego no levantamento e documentação de requisitos para este tipo de sistema e para o ambiente de desenvolvimento existente, promovendo uma melhoria na qualidade das especificações.

REFERÊNCIAS

BOOCH G., RUMBAUGH J., JACOBSON I. *UML guia do usuário*. 2ª Ed. Rio de Janeiro: Campus, 2006.

COCKBURN, Alistair. *Writing effective use cases*. Addison-Wesley Longman, Q3 2000

COCKBURN , Alistair, *Escrevendo casos de uso eficazes*. São Paulo: Bookman, 2001. 254 p.

COHN, Mike, *User Stories Applied*, Addison-Wesley, 2004.

DEDIANA, Mauricio. Histórias de usuários *Metodologias Ágeis de Desenvolvimento*
Disponível em: <http://agilblog.locaweb.com.br/2008/05/29/historias-de-usuario/>
Artigo postado em 29 de maio de 2008 , acessado em 10/04/2011

JEFFRIES, Ron “XProgramming.com : an agile software development resource” postado em 30/08/2001
Disponível em: <http://xprogramming.com/xpmag/expCardConversationConfirmation>
Acessado em 05/04/2011

LARMAN, Craig, *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development*, 3rd. Pearson Education, 2005.

ROBERTSON, James; ROBERTSON, Susane. *Mastering the requirements process*. 2ª Ed. Addison Wesley, 2006.

ROBINSON, William N.; ELOFSON, Greg, *Análise com Casos de Uso baseada em Objetivos*, 2004. Disponível em: http://www.municipioantoniocarlos.mg.gov.br/pos_graduacao_ufrmg/pnt/Analise_com_Casos_de_Uso_baseada_em_ObjetivosR.pdf

SAMPAIO Ana Lúcia de S., PRIMO Francisco F., MARTINO Wagner R. de. *Método para definição de requisitos de software de um sistema a partir das necessidades dos seus stakeholder*. In: SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSOS DE SOFTWARE, 7. São Paulo. Disponível em: http://www.simpros.com.br/upload/A05_2_artigo14647.pdf

SEHLHORST, Scott. *User stories and use cases*, postado em 02/02/2009
Disponível em <http://tynerblain.com/blog/2009/02/02/user-stories-and-use-cases/>
Acessado em 20/05/2011

SEHLHORST, Scott, *Writing complete user story*, postado em 06/07/2009
Disponível em <http://tynerblain.com/blog/2009/07/06/writing-complete-user-stories/>
Acessado em 13/06/2011