



UNIVERSIDADE FEDERAL DE PERNAMBUCO



Centro de Informática

Graduação em Ciência da Computação

Adequação de um processo de reuso de requisitos a
uma empresa desenvolvedora de software

João Henrique Correia Pimentel

Trabalho de Graduação

Recife
2008



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ADEQUAÇÃO DE UM PROCESSO DE REUSO DE REQUISITOS A UMA EMPRESA
DESENVOLVEDORA DE SOFTWARE

João Henrique Correia Pimentel (jhcp@cin.ufpe.br)
Orientadora: Prof. Dra. Carina Frota Alves (cfa@cin.ufpe.br)

Recife
2008



Assinaturas

Este Trabalho de Graduação é resultado da pesquisa realizada pelo aluno João Henrique Correia Pimentel, com a orientação da professora Carina Frota Alves, sob o título de “Adequação de um processo de reuso de requisitos a uma empresa desenvolvedora de software”. Todos abaixo assinados estão de acordo com o conteúdo deste documento e com os resultados deste Trabalho de Graduação.

Carina Frota Alves (orientadora)

João Henrique Correia Pimentel (aluno)

Recife, 24 de janeiro de 2008.

Agradecimentos

Agradeço aos meus pais, Mário e Lourdinha, pelo carinho e apoio que sempre foi uma constante na minha vida. Nunca serei capaz de expressar toda a gratidão que tenho por vocês.

Aos meus irmãos, Ana e Luís, pelo exemplo de dedicação que sempre me deram, e por suportarem as birras deste irmão caçula. Mesmo com a distância física aumentando com o passar do tempo, nossos corações sempre estarão unidos.

À minha namorada, Eline Silva, que vem me alimentando com seu afeto e compreensão, sendo companheira de risos e lágrimas. Sem você, eu teria menos amor no coração e seria mais “adulto”.

À Carina Alves, pela competência e dedicação com que sempre orientou seus trabalhos, e pela paciência com os prazos nem sempre cumpridos... sem você eu teria “desencaminhado” para uma área mais técnica.

Aos colegas do projeto “Um Framework de Engenharia de Requisitos para Desenvolvimento de Produtos de Software”, pelo trabalho nesse que foi o pontapé inicial da presente pesquisa.

Aos amigos que riem e que choram comigo, compartilhando vitórias e derrotas. Em nome de Gustavo e Rodrigo, agradeço a vocês.

Aos integrantes do CITi, por despertar o empreendedorismo e me ensinar coisas que nunca imaginei aprender em uma faculdade. Em nome dos meus colegas de gestão, Allynson, Cesar, Hector, Lúcio e Thiago, agradeço a todos os que passaram pelo CITi.

À minha família, espalhada pelo Brasil, pelos diversos momentos em que estivemos juntos. Em nome de Bruno e Maxwell, primos que vieram fazer a faculdade se hospedando em nossa casa, agradeço a vocês.

Aos colegas da UFPE, que me ajudaram bastante nessa difícil jornada. Em especial, aos colegas do PET, com quem tive a oportunidade de conviver por dois anos, e espero conviver muito mais.

Através do prof. Maurício e de Ismael Northleet, quero agradecer aos colegas da escola de música e da banda Number Nine. Sem a música eu estaria mais perto de um robô do que de um ser humano.

Por último, aos professores e funcionários do Centro de Informática, por fazer deste um ambiente agradável e estimulante.



*"Deixo os versos que escrevi,
As cantigas que cantei,
Cinco ou seis coisas que eu sei
E um milhão que eu esqueci."*

Antônio Carlos Nóbrega e Bráulio Tavares

RESUMO

Este trabalho tem como objetivo definir um processo de reuso de requisitos para uma empresa em específico, realizando transferência tecnológica entre academia e indústria. A partir de uma revisão da literatura, foi identificada a Análise de Domínio como sendo a melhor abordagem para reuso nessa empresa. Realizou-se então um levantamento bibliográfico em Análise de Domínio, sendo escolhida a metodologia *Feature-Oriented Domain Analysis* (FODA) como base do processo.

Foi realizado um estudo de caso, visando a compreensão da empresa no qual o processo seria inserido, utilizando-se as técnicas *Entrevista semi-estruturada*, *Observação* e *Análise de Conteúdo* para a coleta de dados. Essa compreensão foi necessária para que o processo fosse elaborado de acordo com as necessidades específicas dessa organização.

Por último, o processo foi construído e validado em reunião com um representante da empresa. O processo aqui proposto foi intitulado Análise de Domínio Orientada a Características adaptado para a Empresa D (ADOC-D), fazendo referência ao nome fictício adotado para a empresa.

Palavras-chave: Reuso de requisitos, análise de domínio, adaptação de processo.

ABSTRACT

This research primary goal is to define a requirements reuse process for a specific company, enacting technology transfer between academy and industry. Analyzing the literature on requirements reuse, Domain Analysis was identified as being the best approach for reuse in this company. Then, a bibliography survey was made on Domain Analysis, from which the Feature-Oriented Domain Analysis (FODA) methodology was elected as the process basis. A case study was performed, targeting the comprehension of the company on which the process would be inserted, using *Semi-structured interview*, *Observation* and *Content Analysis* as techniques for data gathering. This comprehension was necessary in order to enable the process construction accordingly to the organization specific needs. Finally, the process was developed and validated in a meeting with one company representative. This process was entitled Feature-Oriented Domain Analysis adapted for Company D (in Portuguese, ADOC-D), referring to the company fictional name.

Keywords: Requirements reuse, domain analysis, process adaptation.

SUMÁRIO

1	<i>Introdução</i>	12
1.1	Objetivos	12
1.2	Metodologia	13
1.3	Estrutura do trabalho	14
2	<i>Engenharia de Requisitos</i>	15
2.1	Processo de Engenharia de Requisitos	15
2.1.1	Elicitação	16
2.1.2	Análise e Negociação.....	17
2.1.3	Documentação.....	17
2.1.4	Validação	17
2.1.5	Gerenciamento	17
2.2	Reuso de Requisitos	18
2.3	Considerações finais	19
3	<i>Análise de Domínio</i>	20
3.1	Feature-Oriented Domain Analysis (FODA)	22
3.1.1	Análise de Contexto	23
3.1.2	Modelagem de domínio	25
3.1.3	Modelagem arquitetural	27
3.2	Considerações finais	27
4	<i>Estudo de caso</i>	28
4.1	Metodologia	28
4.1.1	Objetivos.....	29
4.1.2	Plataforma teórica	29
4.1.4	Coleta de dados.....	29
4.1.5	Análise dos dados	30
4.2	Resultados	30
4.2.1	Caracterização da Empresa	31
4.2.1.1	Documentação de ajuda	32
4.2.1.2	Testes	33
4.2.2	Processo de Engenharia de Requisitos na Empresa D	33
4.2.2.1	Desenvolvimento de novas funcionalidades	34
4.2.2.2	Desenvolvimento de novos módulos	35

4.2.2.3 Recuperação de requisitos do legado	36
4.2.2.4 Engenharia de Requisitos Dirigida ao Mercado.....	37
4.2.3 Práticas de Reuso na Empresa D.....	38
4.2.3.1 Potencial de Reuso na Empresa D	39
4.3 Considerações finais	41
5 Processo adaptado de reuso de requisitos.....	42
5.1 Software Process Engineering Metamodel - SPEM.....	43
5.2 Processo - Fases.....	44
5.3.1 Engenharia de Domínio	44
5.3.2 Engenharia de Requisitos.....	45
5.3 Processo - Papéis	46
5.4.1 Especialista do Domínio	46
5.4.2 Analista de Domínio	47
5.4.3 Líder de Equipe.....	48
5.4 Processo - Disciplinas.....	48
5.5.1 Análise de Contexto	48
5.5.2 Modelagem de Domínio.....	51
5.5.3 Requisitos.....	54
5.5 Considerações finais	55
6 Conclusões	56
6.1 Trabalhos futuros	56
<i>Referências bibliográficas.....</i>	<i>58</i>
<i>Apêndice A – Template de Glossário</i>	<i>64</i>
<i>Apêndice B – Template de Modelo de Contexto</i>	<i>67</i>
<i>Apêndice C – Template de Modelo de Contexto</i>	<i>71</i>
<i>Anexo A - Questionário aplicado na Empresa D em estudo prévio</i>	<i>75</i>
<i>Anexo B - Roteiro da entrevista realizada na Empresa D em estudo prévio.....</i>	<i>85</i>

LISTA DE FIGURAS

Figura 1 – Visão geral da metodologia utilizada.....	13
Figura 2 - Processo de engenharia de requisitos no modelo espiral.....	16
Figura 3- Visão geral de gerenciamento de requisitos. Fonte: (Alves 2006 b)	18
Figura 4- Diagrama de contexto do processo de Análise de Domínio. Adaptado de (Prieto-Diaz 1990)	21
Figura 5 - Relação entre os produtos da análise de domínio e os produtos do desenvolvimento de um novo sistema. Adaptado de (Kang 1990).....	23
Figura 6 - Diagrama de estrutura no domínio de gerenciamento de janelas. Adaptado de (Kang 1990).....	24
Figura 7 - Diagrama de fluxo de dados no domínio de gerenciamento de janelas. Adaptado de (Kang 1990).....	25
Figura 8 – Exemplo de modelo de características do FODA. Adaptado de (Kang 1990).....	26
Figura 9 – Visão geral da metodologia de estudo de caso.....	29
Figura 10 – Estereótipos do SPEM utilizados em ADOC-D.....	44
Figura 11 – Fases do processo ADOC-D	44
Figura 12 – Disciplinas da fase Engenharia de Domínio	45
Figura 13 – Disciplina da fase Engenharia de Requisitos	45
Figura 14 – Papéis do processo ADOC-D.....	46
Figura 15 – Diagrama de casos de uso do papel Especialista do Domínio	47
Figura 16 – Diagrama de casos de uso do papel Analista de Domínio	47
Figura 17 – Diagrama de casos de uso do papel Líder de Equipe.....	48
Figura 18 – Diagrama de pacotes da disciplina Análise de Contexto	49
Figura 19 – Diagrama de atividades da disciplina Análise de Contexto.....	50
Figura 20 – Diagrama de pacotes da disciplina Modelagem de Domínio.....	52
Figura 21 – Diagrama de atividades da disciplina Modelagem de Domínio	53
Figura 22 – Diagrama de pacote da disciplina Requisitos.....	54

LISTA DE TABELAS

Tabela 1 - Atividades de Análise de Domínio. Adaptado de (Arango 1994).....	21
Tabela 2 - Entradas, atividades e resultados das fases de FODA.....	22
Tabela 3 - Características da Empresa D.....	31
Tabela 4 - Confrontamento da situação da Empresa D com hipóteses sobre Engenharia de Requisitos Dirigida ao Mercado	37
Tabela 5 - Avaliação do produto da Empresa D de acordo com a definição de COTS	38
Tabela 6 - Situação da Empresa D em relação aos fatores de sucesso de reuso	40

1 Introdução

Diversos estudos mostram que o processo de Engenharia de Requisitos (ER) é um fator crítico para o sucesso de projetos de *software*. De acordo com (Gartner 1995), aproximadamente 60-70% das falhas ocorridas em projetos de desenvolvimento de sistemas de *software* ocorrem devido a inadequação do processo de aquisição, análise e gerenciamento de requisitos. O reuso de requisitos é uma prática que pode diminuir a quantidade dessas falhas e a quantidade de recursos utilizados para a elicitacão e a documentacão de requisitos, especialmente no desenvolvimento de sistemas de domínio específico.

De acordo com a literatura (Lam 1998 a) (Lucrédio 2007), o reuso de requisitos só gera bons resultados se aplicado através de um processo sistemático, e não através de práticas *ad-hoc*. Apesar de o estado-da-arte no assunto se encontrar bastante avançado, se observa pouca adocão de reuso nas empresas. Em um estudo empírico recentemente realizado com 13 empresas desenvolvedoras de software pernambucanas (Alves 2007) (Alves 2006 b), se observou que nenhuma delas adotava sistematicamente algum processo ou prática de reuso de requisitos.

Dentre os fatores observados na literatura como obstáculos para adocão de reuso nas empresas desenvolvedoras, estão a magnitude do esforço de adocão e a existênciade pouca transferênciatecnológica na área (Almeida 2007) (Lam 1998 b).

1.1 Objetivos

Este trabalho tem como objetivo apresentar um processo adaptado de reuso de requisitos para adocão em uma empresa desenvolvedora de software, fazendo transferênciatecnológica entre a academia e a indústria. Para se atingir este objetivo foi realizado um estudo da literatura sobre o assunto, um estudo de caso para análise do ambiente organizacional em questão e a adaptacão de um processo de reuso de requisitos a partir da metodologia *Feature-Oriented Domain Analysis* (FODA) (Kang 1990). Esse estudo de caso se fez necessário para que o processo adaptado pudesse realmente satisfazer as necessidades da empresa, levando em consideracão suas peculiaridades.

Para a realizacão deste estudo, foi selecionada uma empresa na qual havia uma grande demanda por reuso de requisitos, um forte interesse na realizacão do estudo e cujos sistemas desenvolvidos apresentavam uma complexidade e similaridade que sugeria consideráveis

benefícios na adoção de reuso de requisitos. Essa empresa vem observando a existência de desperdício de recursos oriunda da não-adoção de reuso em nenhuma etapa do desenvolvimento de *software*, e enxerga o reuso de requisitos como a melhor forma de se iniciar a inserção de uma sistemática de reuso no seu ciclo de desenvolvimento.

A existência de um contato preliminar já estabelecido com a empresa, onde inclusive o pesquisador já realizou uma pesquisa junto à empresa, facilitou a interação entre as partes. Essa pesquisa realizada foi relatada em (Alves 2007), e as informações previamente obtidas auxiliaram este estudo na compreensão dos esforços da empresa em requisitos. Dessa forma, pode-se considerar que o presente trabalho é uma continuação de pesquisa anterior.

1.2 Metodologia

Este projeto seguiu a metodologia apresentada na Figura 1. Inicialmente foi realizado o planejamento da pesquisa, sendo necessário uma reunião com a empresa a ser estudada para se chegar a um acordo sobre o projeto. Esse planejamento levou em conta os relatos sobre transferência tecnológica apresentados em (Lam 1998 b) (Pinheiro 2003), em particular no que se refere ao comprometimento da alta direção e à especificidade do projeto.

O passo seguinte foi uma revisão bibliográfica, buscando adquirir conhecimento sobre reuso de requisitos, que embasaria o trabalho. A partir da decisão de se utilizar a metodologia *Feature-Oriented Domain Analysis* (FODA) na construção do processo, fez-se um aprofundamento na literatura sobre essa metodologia.

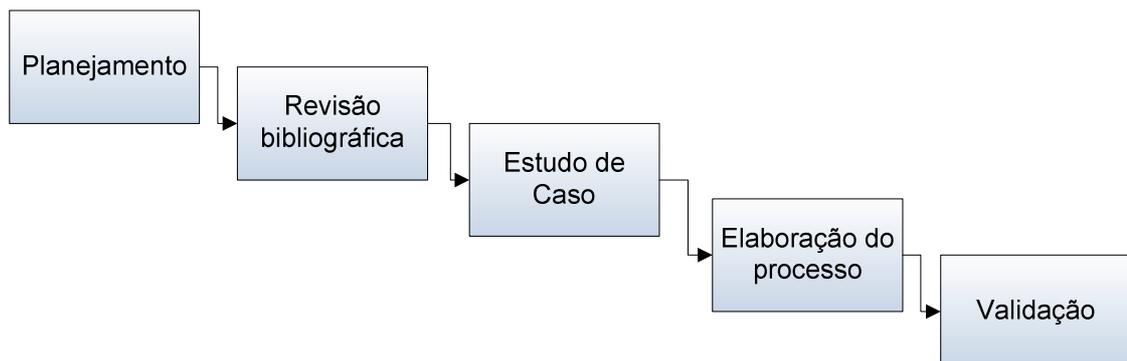


Figura 1 – Visão geral da metodologia utilizada

Uma vez realizada a revisão da literatura, iniciou-se um estudo de caso, onde se pretendia compreender o contexto no qual o processo proposto seria inserido. A metodologia utilizada para a realização do estudo de caso, em particular, é apresentada no capítulo 4.

A partir das informações obtidas nas etapas anteriores foi elaborado um processo de reuso de requisitos, específico para a empresa no qual foi realizado o estudo de caso. Esse processo foi então validado com a empresa, sendo submetido a alguns ajustes e finalizado.

1.3 Estrutura do trabalho

Este trabalho está dividido em seis capítulos, três apêndices e dois anexos. Neste primeiro capítulo é feita uma introdução sobre o trabalho, descrevendo seus objetivos e sua metodologia.

No capítulo 2 são apresentados conceitos básicos de Engenharia de Requisitos, sendo apresentado um processo base de Engenharia de Requisitos. Também é apresentada uma visão geral sobre reuso de *software*, com ênfase em reuso de requisitos.

Em seguida, o capítulo 3 apresenta os conceitos de Análise de Domínio, introduzindo seus conceitos básicos e detalhando a metodologia *Feature-Oriented Domain Analysis* (FODA), com suas fases e atividades.

O capítulo 4 apresenta o estudo de caso realizado, descrevendo a metodologia utilizada e os resultados encontrados. Os resultados são divididos em *Caracterização da Empresa*, *Processo de Engenharia de Requisitos na Empresa D* e *Práticas de Reuso na Empresa D*.

O processo definido neste trabalho é apresentado no capítulo 5. Para descrevê-lo foi utilizado o metamodelo *Software Process Engineering Metamodel* – SPEM, que é brevemente apresentado nesse capítulo.

O sexto capítulo expõe as conclusões do trabalho, apontando elementos de destaque no texto e indicando trabalhos futuros que podem ser realizados. Os apêndices apresentam modelos de documentos (*templates*) que fazem parte do processo proposto, enquanto os anexos apresentam instrumentos de coleta de dados utilizados na pesquisa anterior.



2 Engenharia de Requisitos

Este capítulo apresenta uma visão geral do processo de engenharia de requisitos, com suas fases e características. Em seguida serão discutidos alguns conceitos básicos de reuso de requisitos.

Engenharia de requisitos se preocupa com a identificação de metas a serem atingidas pelo sistema a ser desenvolvidos, assim como seus serviços e restrições (Zave 1997). Mais especificamente, é na engenharia de requisitos que se descobre, analisa, documenta e verifica os requisitos (Sommerville 2003). Os requisitos podem se originar de qualquer pessoa ou organização que tenha algum interesse no projeto a ser desenvolvido, chamados *stakeholders* (Robertson 2000).

Requisitos, por sua vez, descrevem o que o sistema faz para o usuário e quais restrições ele precisa satisfazer (Robertson 2006). Os requisitos podem ser divididos em requisitos do usuário, que descrevem o que o usuário espera do sistema, e requisitos do sistema, que descrevem o que o sistema faz. Eles também podem ser classificados como requisitos funcionais, que são as ações que o sistema deve fazer, e requisitos não-funcionais, que descrevem as propriedades e qualidades que o sistema deve ter (Sommerville 2003). Os requisitos não-funcionais são muitas vezes chamados requisitos de qualidade.

2.1 Processo de Engenharia de Requisitos

De acordo com (Sommerville 2003), o processo de engenharia de requisitos é o processo que envolve todas as atividades exigidas para se criar e manter o documento de requisitos do sistema.

Para este trabalho adotaremos como base o processo proposto por (Kotonya 1997), segundo o qual o processo de engenharia de requisitos compreende as fases de elicitação, análise e negociação, documentação e validação de requisitos. A Figura 2 apresenta o processo no modelo espiral, composto por atividades iterativas, inter-relacionadas e retro-alimentadas (Nuseibeh 2000). Além dessas quatro atividades, diversos autores também falam sobre a atividade Gerenciamento de Requisitos (Rajat 2003) (Sommerville 2003).

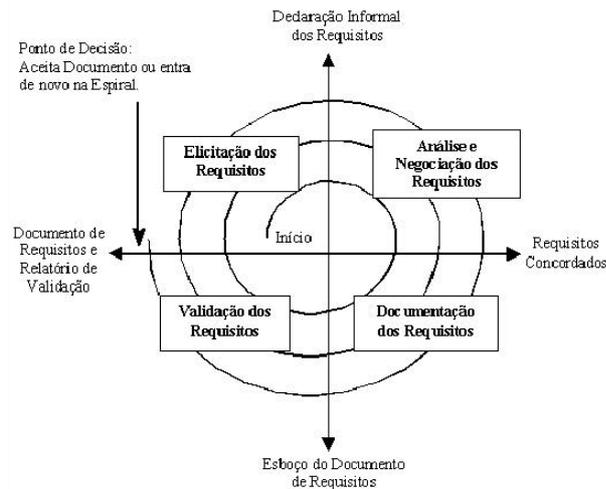


Figura 2 - Processo de engenharia de requisitos no modelo espiral

2.1.1 Elicitação

Esta é a atividade na qual, através de diversas técnicas, se identificam os requisitos dos diversos *stakeholders*. O objetivo é o entendimento correto e completo do que se espera do sistema sendo desenvolvido. Em geral a elicitação se baseia na comunicação entre os *stakeholders* do projeto.

A escolha da técnica de elicitação deve ser feita caso a caso, a depender do contexto no qual o projeto está inserido. As técnicas são classificadas em

- Técnicas tradicionais – as técnicas de uso comum, como aplicação de questionários, observação e análise de documentos.
- Técnicas de elicitação de grupo – analisam o comportamento e a dinâmica dos grupos. Exemplos são *brainstorming*, *Joint Application Design (JAD)* e *Rapid Application Development (RAD)*.
- Técnicas cognitivas – uso de técnicas de aquisição de conhecimento para a elicitação de requisitos. Exemplos incluem análise de protocolo, *laddering*, *card sorting* e *repertory grids*.
- Técnicas contextuais – se baseiam no contexto no qual o sistema será inserido. Exemplos são etnografia e análise social.
- Técnicas de modelagem – se baseiam na utilização de um modelo específico que descreve as informações adquiridas. Exemplos são *Knowledge Acquisition in autOmated Specification (KAOS)*, *i**, e casos de uso.
- Prototipação – utiliza uma versão inicial do sistema para refinar a elicitação. É utilizada quando há um alto grau de incerteza no projeto.

2.1.2 Análise e Negociação

Nesta atividade se procura fornecer descrições abstratas e facilmente interpretáveis dos requisitos. A descrição dos requisitos costuma incluir sua modelagem, propiciando uma representação visual padronizada. A modelagem pode ser organizacional, comportamental, de domínio ou de requisitos não-funcionais, entre outros.

Durante a análise, eventualmente serão identificados problemas com os requisitos, como conflitos entre requisitos, ambigüidade, duplicação e incompletude. Para se resolver esses problemas são necessários discussão, priorização e negociação dos mesmos entre os *stakeholders*. A negociação pode ser conduzida *ad-hoc* ou utilizando técnicas específicas como *Win-Win* e *Analytic Hierarchy Process (AHP)*.

2.1.3 Documentação

Nesta atividade são documentados os requisitos do sistema, seja em documento ou em um banco de dados através de uma ferramenta de requisitos. Os requisitos podem ser descritos em linguagem natural ou em linguagens lógicas e formais. A norma IEEE/ANSI 830-1998 sugere que o documento de requisitos deva ser não-ambíguo, completo, verificável, consistente, modificável, rastreável e utilizável durante todas as fases do ciclo de vida do requisito.

2.1.4 Validação

Esta é a atividade na qual se certifica que o documento de requisitos é consistente com as necessidades dos *stakeholders* (Loucopoulos 1995). Ao final desta atividade espera-se que exista um entendimento único e em comum acordo entre o cliente e a equipe de desenvolvimento sobre o que será desenvolvido. Entre os critérios a serem considerados nesta atividade, estão a validade dos requisitos, a consistência, a completude, a viabilidade e a facilidade de verificação (Sommerville 2003).

2.1.5 Gerenciamento

O gerenciamento de requisitos é uma atividade transversal às demais, que objetiva controlar a evolução dos requisitos. Essa evolução pode decorrer da constatação de novas necessidades dos *stakeholders* ou da constatação de deficiência nos requisitos já registrados (Thayer 1997). De acordo com (Kotonya 1997), as principais preocupações do gerenciamento de requisitos são gerenciar mudanças nos requisitos acordados, gerenciar os relacionamentos

entre os requisitos e gerenciar as dependências entre o documento de requisitos e outros documentos produzidos ao longo do processo.

A Figura 3 ilustra os componentes básicos do gerenciamento de requisitos. Todas essas atividades estão ligadas aos requisitos em si. O controle de mudanças deve facilitar a evolução dos requisitos e permitir que nenhuma solicitação de mudança seja perdida ou desconsiderada. O gerenciamento de configuração trata das diferentes versões dos artefatos, incluindo suas revisões, bem como dos diferentes *releases* do sistema em desenvolvimento. Rastreabilidade é a identificação, manutenção e visualização dos relacionamentos entre requisitos (rastreabilidade vertical), e entre os requisitos e demais elementos do sistema, como classes e casos de teste (rastreabilidade horizontal). Gerenciar qualidade, neste contexto, é adotar mecanismos que permitam um menor re-trabalho por falhas nos requisitos.



Figura 3- Visão geral de gerenciamento de requisitos. Fonte: (Alves 2006 b)

2.2 Reuso de Requisitos

De acordo com (Basili 1991), o reuso de software é o reuso de tudo associado a um projeto de software, inclusive conhecimento. Tem como objetivo aumentar a qualidade e a produtividade do desenvolvimento de *software* (Almeida 2007). Em particular, o reuso de requisitos tem um impacto positivo em todo o ciclo de desenvolvimento (Cybulski 1998).

O reuso de requisitos é um processo onde se faz o uso sistemático de documentos de requisitos visando reduzir o esforço dentro do ciclo de vida do *software* (Villegas 2001). O reuso desses artefatos iniciais do ciclo de desenvolvimento, chamado de *early reuse* na literatura, não só melhora o processo de engenharia de requisitos como também apóia o desenvolvimento com reuso em fases posteriores (Villegas 2001) (Lam 1998 b).

Os requisitos podem ser classificados, quanto ao reuso, em (Keepence 1995):

- Não reusável – Requisitos que estão diretamente relacionados à aplicação sendo desenvolvida são possivelmente não-reusáveis.

- Diretamente reusável – Requisitos que podem ser inteiramente reutilizados. Normalmente são atrelados a um domínio específico ou são oriundos de padrões e normas.
- Parametrizado ou Templatizado – Requisitos que podem ser reutilizados com a variação de algum dos seus elementos, que são parametrizados em vez de fixos.

Para a implantação de um processo de reuso, é importante que se defina métricas para avaliar os resultados da implantação. As métricas de reuso podem ser classificadas como *Economics Oriented Reuse Metrics and Models* (EORM), *Software Structure Oriented Reuse Metrics* (SORM) e *Reuse Repository Metrics* (RRM) (Almeida 2007). EORM avalia o impacto da implantação de reuso nas organizações, associados a retorno sobre o investimento. SORM avalia a implantação pelo lado técnico, avaliando o que e como está sendo reusado. RRM avalia características do repositório de reuso utilizado, como disponibilidade de serviço e quantidade de recursos disponíveis. Exemplos de métricas EORM são Custo Evitado pelo Reuso, Valor Agregado pelo Reuso e Retorno Sobre Investimento. Em SORM temos Nível de Reuso, Frequência de Reuso, Taxa de Reuso e Densidade de Reuso, entre outros. Métricas RRM podem incluir Taxa de Acesso, Quantidade de Artefatos e Taxa de Disponibilidade.

Dentre as diferentes formas de reuso de requisitos, um ponto central na maioria delas é o conceito de domínio. Acredita-se que aplicações dentro de um mesmo domínio possuem grande probabilidade de possuírem requisitos similares. Este tema será aprofundado no próximo capítulo.

2.3 Considerações finais

Na seção 2.1 deste capítulo foram apresentados alguns conceitos básicos do processo de Engenharia de Requisitos e as principais atividades que o compõem: Elicitação, Análise e Negociação, Documentação, Validação e Gerenciamento.

Em seguida, na seção 2.2, foram apresentados os principais conceitos de reuso de requisitos, com uma breve introdução sobre reuso de *software* em geral.

No capítulo seguinte serão apresentados os conceitos de análise de domínio e como sua utilização pode favorecer o reuso de requisitos. Em particular, será apresentada a metodologia *Feature-Oriented Domain Analysis* (FODA), na qual o processo proposto neste trabalho se baseia.



3 Análise de Domínio

Neste capítulo serão apresentados conceitos básicos de Análise de Domínio e as características genéricas de um processo de Análise de Domínio, para em seguida se apresentar a metodologia *Feature-Oriented Domain Analysis* (FODA).

Do ponto de vista de *software*, domínio é um grupo de sistemas, ou de áreas funcionais dentro de um sistema, que exibem funcionalidades similares (SEI 2007). Quando o domínio abrange um sistema inteiro, como o domínio de sistemas de prontuários eletrônicos, é chamado domínio vertical. O domínio de partes de um sistema, como o domínio de sistemas de banco de dados, é conhecido como domínio horizontal (Czarnecky 2000).

O termo Análise de Domínio foi introduzido na literatura por (Neighbors 1980), como sendo a atividade de identificar os objetos e ações de um grupo de sistemas em um domínio particular. Esta é uma das atividades da Engenharia de Domínio, que além dela também compreende o desenvolvimento da arquitetura e a implementação dos componentes (Czarnecki 2000). Para que se realize a análise de um domínio, é essencial que ele seja maduro, razoavelmente estável e economicamente viável (Simos 1995). Esses critérios visam garantir que a análise gere um bom retorno para a organização que a realiza.

Uma definição mais moderna para Análise de Domínio considera que este é um processo pelo qual informações utilizadas no desenvolvimento de sistemas para um domínio são identificadas, capturadas e organizadas visando torná-las reusáveis (Prieto-Diaz 1990). Esse mesmo trabalho apresenta um diagrama de contexto desse processo, ilustrado na Figura 4. Como entrada do processo temos informações obtidas na literatura, em artefatos de sistemas já existentes, em pesquisas de mercado, em requisitos de sistemas e em especialistas, que são as fontes de conhecimento do domínio. Métodos e técnicas de Análise de Domínio e de gerenciamento são aplicados por especialistas no domínio do problema e por analistas de domínio, com apoio de engenheiros de domínio. Como resultado final temos o modelo de domínio, que pode compreender sua taxonomia, seus padrões, seus modelos funcionais e sua linguagem.

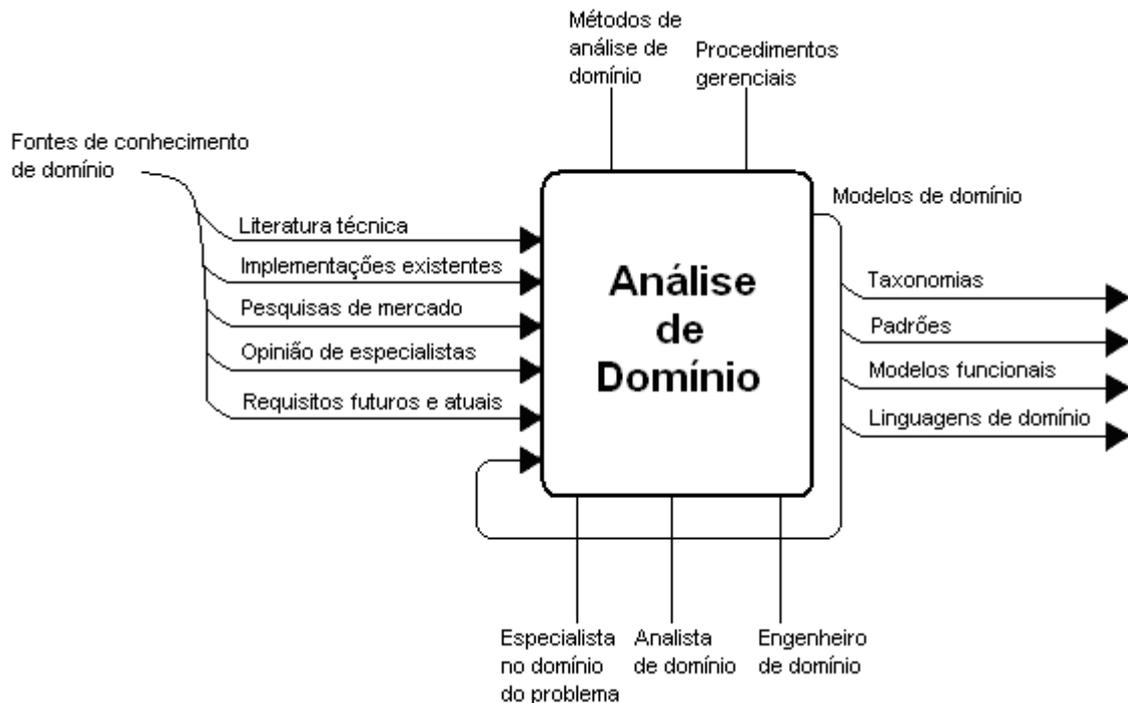


Figura 4- Diagrama de contexto do processo de Análise de Domínio. Adaptado de (Prieto-Diaz 1990)

Uma lista de atividades de Análise de Domínio, compilada a partir de 8 diferentes métodos de Análise de Domínio, é apresentada em (Arango 1994). Essa lista está reproduzida na Tabela 1.

Tabela 1 - Atividades de Análise de Domínio. Adaptado de (Arango 1994)

Principais componentes do processo de Análise de Domínio	Atividades de Análise de Domínio
Caracterização do domínio (planejamento e escopo do domínio)	Seleção do domínio – Realizar análise de negócios e de riscos para determinar qual domínio atende os objetivos de negócio da organização.
	Descrição do domínio – Define as fronteiras e o conteúdo do domínio.
	Identificação da fonte de dados – Identificar as fontes de conhecimento sobre o domínio.
	Preparação de inventário – Criar um inventário de fonte de dados.
Levantamento de dados (modelagem do domínio)	Recuperação de abstrações – Recuperar abstrações.
	Elicitação de conhecimento – Elicitar conhecimento de especialistas.
	Revisão da literatura.
	Análise de contexto e de cenários.
Análise de dados (modelagem de domínio)	Identificação de entidades, operações e relacionamentos.
	Modularização – Utilizar alguma técnica de modelagem adequada.
	Análise de similaridade – Analisar as similaridades entre entidades, atividades, eventos, relacionamentos, estruturas, etc.
	Análise de variações - Analisar as variações entre entidades, atividades, eventos, relacionamentos, estruturas, etc.
	Análise de combinações – Analisar combinações que sugerem estruturas típicas ou padrões de comportamento.
	Análise de balanceamento – Analisar balanceamentos que sugerem possíveis decomposições de módulos e arquiteturas para satisfazer conjuntos incompatíveis de requisitos encontrados no domínio.
Classificação taxonômica	Agrupamento – Agrupar as descrições.

(modelagem do domínio)	Abstração – Criar abstrações das descrições.
	Classificação – Classificar as descrições.
	Generalização – Generalizar as descrições.
	Construção de vocabulário.
Avaliação	Avaliar o modelo de domínio.

Para definir a metodologia de Análise de Domínio a ser usada como base do presente trabalho, foram analisadas 8 metodologias, a saber: DADO (Maccario 1997), Draco (Freeman 1987), FODA (Cohen 1992), HP (Cornwell 1996), IdeA (Lubars 1989), JODA (Holibaugh 1992), ODM (Simos 1995) e STARS (Creps 1992). As pesquisas apresentadas em (Ferré 1999) e (Ahmad 2004) auxiliaram na comparação das metodologias.

A partir da análise das metodologias de Análise de Domínio, foi escolhida a metodologia chamada *Feature-Oriented Domain Analysis* (FODA) (Cohen 1992) (Kang 1990), livremente traduzida como Análise de Domínio Orientada a Características. Os principais fatores que levaram à escolha da metodologia FODA foram:

- Boa documentação;
- Facilidade de assimilação dos seus conceitos e
- Suporte a reuso em demais etapas do desenvolvimento de *software*.

3.1 Feature-Oriented Domain Analysis (FODA)

A metodologia FODA se originou de um estudo de diferentes abordagens de Análise de Domínio, e provê o reuso não só de requisitos como também de arquitetura. O foco deste método é a identificação das características proeminentes dos *softwares* de um domínio, que definem tanto as similaridades quanto as diferenças desses *softwares*.

FODA se divide em três fases: Análise de Contexto, Modelagem de Domínio e Modelagem de Arquitetura. As entradas, as atividades e os resultados de cada fase são apresentados na Tabela 2.

Tabela 2 - Entradas, atividades e resultados das fases de FODA

Fase	Entradas	Atividades	Resultados
Análise de Contexto	Ambiente de operação; Padrões	Análise de contexto	Modelo de contexto
Modelagem de Domínio	Características; Modelo de contexto	Análise das características	Modelo de características
	Conhecimento sobre o domínio	Modelagem entidade-relacional	Modelo entidade-relacional
	Tecnologia do domínio; Modelo de contexto; Modelo de características; Modelo entidade-relacional; Requisitos	Análise funcional	Modelo funcional Modelo comportamental

Modelagem arquitetural	Tecnologia de implementação; Modelo de contexto; Modelo de características; Modelo entidade-relacional; Informações de design	Modelagem arquitetural	Modelo de arquitetura
------------------------	---	------------------------	-----------------------

A Figura 5 apresenta a relação entre os produtos da análise de domínio – desenvolvimento *para* reuso – e os produtos do desenvolvimento de um novo sistema – desenvolvimento *com* reuso. O processo de análise de domínio é retro-alimentado durante o desenvolvimento de um novo sistema, passando pela análise de um especialista e de um analista do domínio. Por desenvolvimento *para* reuso entende-se todo o esforço que é depreendido visando possibilitar o reuso do conhecimento que está sendo gerado, enquanto desenvolvimento *com* reuso é a utilização do conhecimento gerado anteriormente com uma finalidade única e específica.

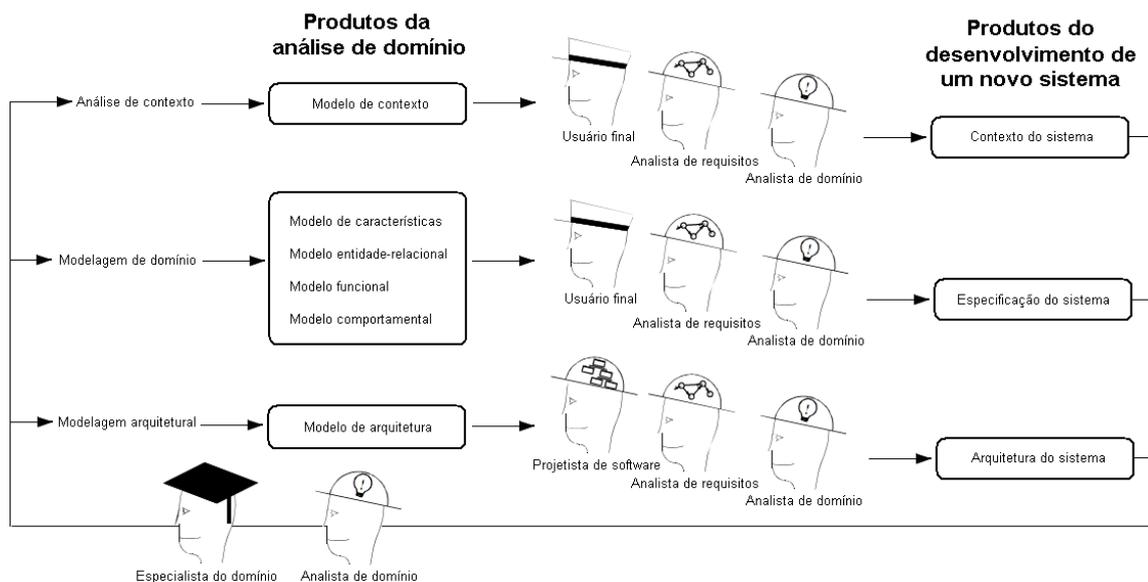


Figura 5 - Relação entre os produtos da análise de domínio e os produtos do desenvolvimento de um novo sistema. Adaptado de (Kang 1990)

A seguir serão apresentadas cada uma das fases de FODA.

3.1.1 Análise de Contexto

Nesta fase se define o escopo do domínio a ser explorado. São analisados os relacionamentos entre o domínio e elementos externos, e as possíveis variações são avaliadas. O resultado final desta fase é o documento de modelo de contexto, que define as fronteiras do domínio.

O modelo de contexto é constituído por um ou mais diagramas de estrutura e de fluxo de dados. O diagrama de estrutura é um diagrama de blocos informal, no qual o domínio a ser analisado é posicionado relativamente a outros domínios com os quais está relacionado, inclusive eventuais subdomínios do domínio em questão. Se necessário, pode ser utilizado mais de um diagrama desse tipo. A Figura 6 exemplifica um diagrama de estrutura com o domínio de gerenciamento de janelas (retângulo em destaque).

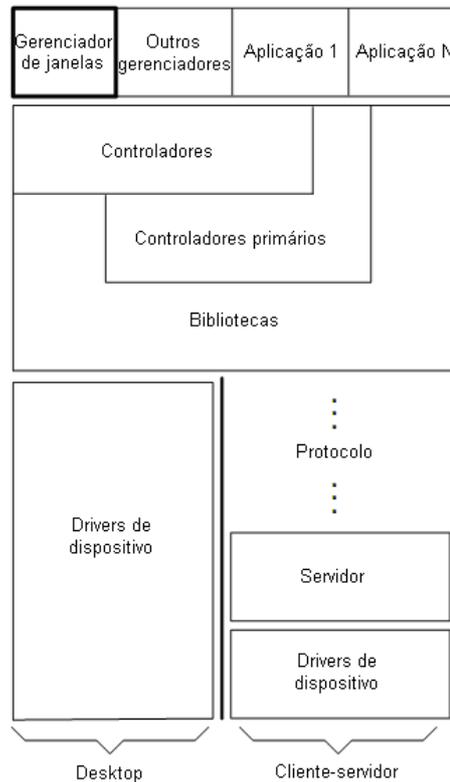


Figura 6 - Diagrama de estrutura no domínio de gerenciamento de janelas. Adaptado de (Kang 1990)

O diagrama de fluxo de dados apresenta o fluxo de dados entre o domínio em análise e todos os outros domínios e entidades com os quais se comunica. Esse diagrama deve ser abstrato o suficiente para abranger as diferentes aplicações que fazem parte do domínio. A Figura 7 exemplifica um diagrama de fluxo de dados com o domínio de gerenciamento de janelas, representado por um círculo.

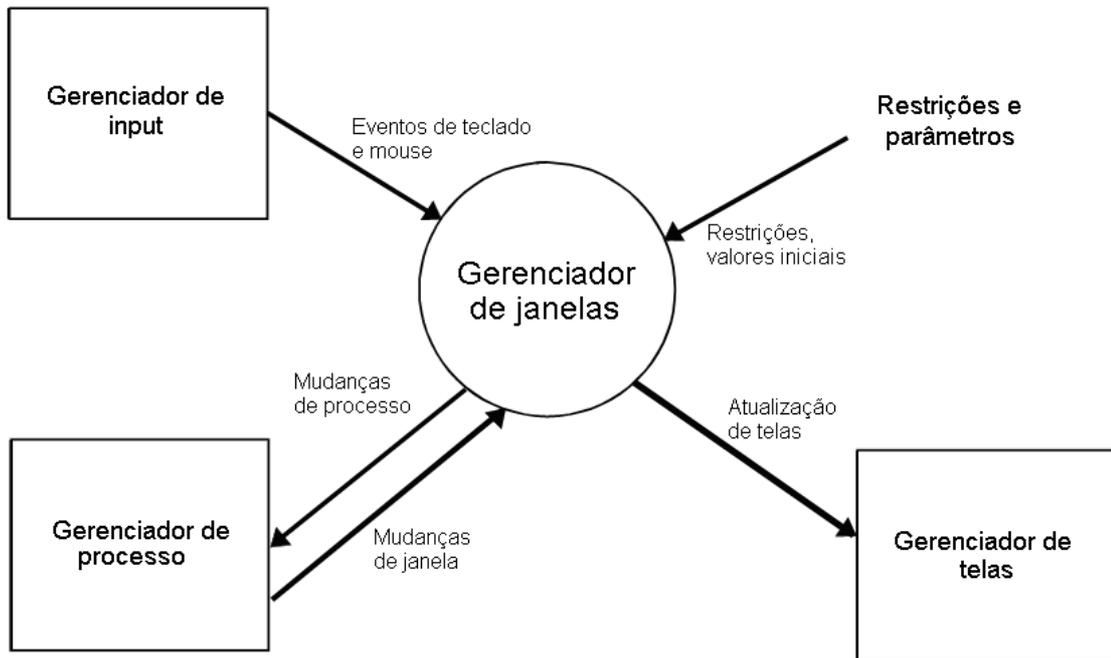


Figura 7 - Diagrama de fluxo de dados no domínio de gerenciamento de janelas. Adaptado de (Kang 1990)

3.1.2 Modelagem de domínio

Nesta fase se identifica as similaridades e as diferenças que caracterizam as aplicações do domínio. Esta fase possui três atividades principais:

Análise de características – nesta atividade se identifica as características dos sistemas que são visíveis ao consumidor ou usuário final. As variações das características são apresentadas em uma decomposição hierárquica de características indicando quais são opcionais ou alternativas. Além disso, o modelo de características documenta as definições das características e as regras de composição. A Figura 8 ilustra a decomposição hierárquica para o domínio de carros. Segundo esse exemplo, um carro obrigatoriamente possui transmissão e potência, podendo ou não ter ar-condicionado. Por sua vez a transmissão pode ser, exclusivamente, manual ou automática.

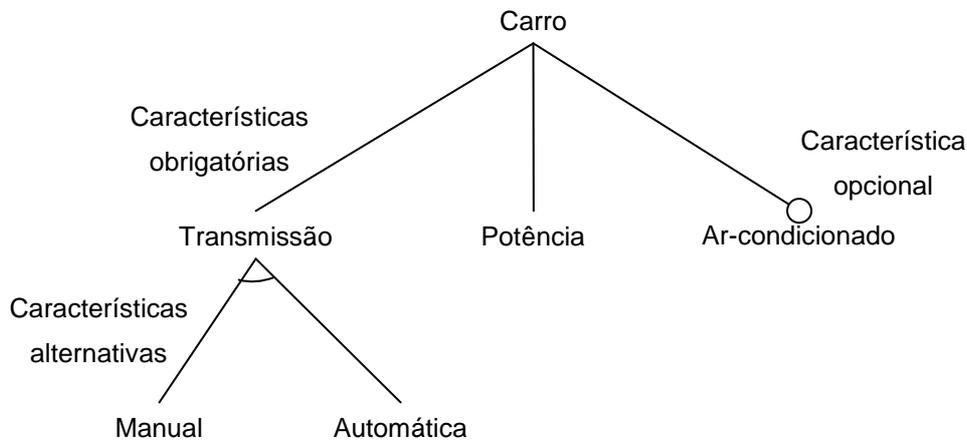


Figura 8 – Exemplo de modelo de características do FODA. Adaptado de (Kang 1990)

Cada característica deve possuir um nome único, e sua definição deve constar em um dicionário de terminologia do domínio. As características alternativas podem ser consideradas uma especialização (ex: transmissão manual é uma especialização de transmissão), por isso a definição de uma característica é herdada pelas suas alternativas.

As regras de composição definem a semântica das relações entre características que não são expressas na decomposição hierárquica. Todas as características opcionais e alternativas (A) que não podem ser usadas junto a outras (B) devem ser relacionadas como A “mutuamente excludente a” B. As características opcionais e alternativas (C) que devem ser usadas em conjunto a outras (D), devem ser relacionadas como C “requer” D.

Modelagem entidade-relacional – nesta atividade se identifica e define o conhecimento de domínio que é essencial para a implementação de aplicações nesse domínio. No desenvolvimento de aplicações sem análise de domínio, essas informações costumam ficar apenas escondidas no código.

O propósito desta modelagem é representar o conhecimento sobre o domínio explicitamente em termos de entidades de domínio e seus relacionamentos, facilitando a derivação para a definição de dados e objetos durante a análise funcional e a modelagem da arquitetura.

A representação gráfica do modelo é baseada em (Chen 1976), sendo formada basicamente por entidades e relacionamentos do tipo “consiste de” (agregação) e “é um” (generalização). Outros relacionamentos podem ser utilizados, contanto que suas definições

sejam devidamente documentadas. As entidades são unidades de informação do domínio que devem ser processadas ou armazenadas pelos sistemas.

Análise funcional – nesta atividade se identifica as funcionalidades em comum e diferenças funcionais das aplicações do domínio. São criadas abstrações e estruturas em um modelo a partir do qual especificações funcionais de aplicações específicas podem ser derivadas.

Os resultados desta atividade são os modelos funcionais e os modelos comportamentais. O modelo funcional descreve os aspectos estruturais das aplicações em termos de entradas, saídas, atividades, dados internos, estruturas lógicas e relacionamentos. O modelo comportamental descreve como uma aplicação se comporta em termos de eventos, entradas, estados, condições e transições de estados.

3.1.3 Modelagem arquitetural

Nesta fase é gerado um modelo de arquitetura, ou projeto de referência, a partir do qual um projeto detalhado e a construção de componentes podem ser feitos. A arquitetura é dividida em camadas, visando facilitar o reuso. O foco nesta modelagem é identificar processos concorrentes e módulos, aos quais serão alocadas as características, as funções, e os dados definidos no modelo de domínio.

Como o escopo deste trabalho não abrange a definição de arquitetura, esta fase não será devidamente detalhada.

3.2 Considerações finais

No início deste capítulo foi apresentada uma visão geral sobre Análise de Domínio, com seus conceitos básicos e o modelo de um processo genérico de Análise de Domínio.

Em seguida, na seção 3.1, foi apresentada a metodologia *Feature-Oriented Domain Analysis* (FODA), na qual este trabalho se baseia. Foram apresentadas as fases que a compõem, com suas respectivas atividades.

No próximo capítulo será apresentado o estudo de caso conduzido neste trabalho. Com esse estudo de caso foi possível obter informações sobre a empresa à qual seria proposto um processo de reuso de requisitos, adaptando-o à sua realidade.



4 Estudo de caso

Neste capítulo será apresentado o estudo de caso realizado visando obter informações sobre a empresa para a qual o processo de reuso de requisitos foi adaptado. Inicialmente será descrita a metodologia empregada, e em seguida os resultados do estudo de caso, separados em Caracterização da Empresa, Processo de Engenharia de Requisitos e Práticas de Reuso.

4.1 Metodologia

Este estudo foi elaborado seguindo a metodologia para condução de estudo de caso descrita em (Martins 2006). Inicialmente foram definidos os objetivos do estudo e elaborada a plataforma teórica. Em seguida, os dados foram coletados, através das seguintes técnicas:

- *Entrevista semi-estruturada* – técnica onde o pesquisador elabora perguntas guias para a entrevista mas não tem a obrigação de seguir um roteiro fixo, podendo fazer mais questionamentos que lhe ocorrerem durante a realização da entrevista. Também não há restrições quanto às respostas do entrevistado;
- *Observação* – nessa técnica o pesquisador observa o fenômeno a ser estudado, fazendo suas inferências a partir do que é observado. Pode ser classificada em cinco dimensões: secreta ou pública, não-participante ou participante, sistemática ou não-sistemática, em situação natural ou em situação artificial e auto-observação ou observação de outros;
- *Análise de Conteúdo* – técnica na qual o pesquisador analisa documentos relacionados ao seu objeto de estudo, podendo realizar suas próprias inferências sobre o conteúdo ali presente.

Os dados obtidos foram documentados em um relatório discursivo, que foi validado pelos participantes do estudo. Por último foi realizada a análise do estudo, de modo empírico. A Figura 9 apresenta uma visão geral da metodologia empregada neste estudo de caso. Nas próximas seções são apresentadas informações adicionais sobre cada passo executado.

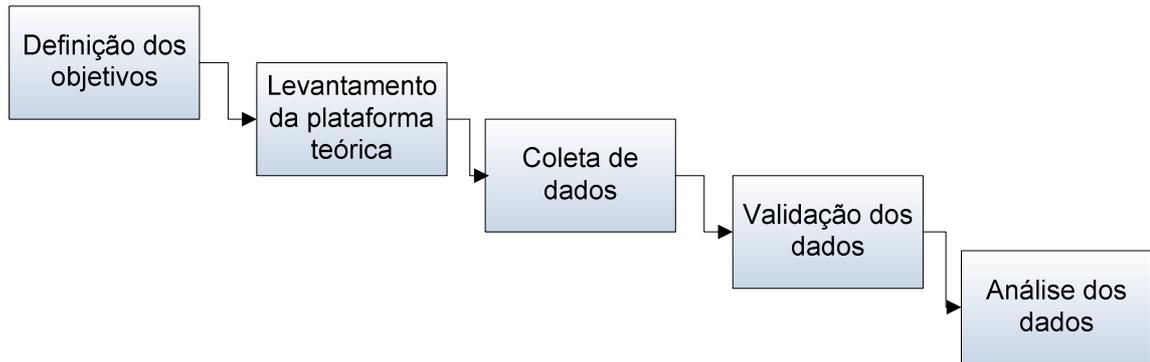


Figura 9 – Visão geral da metodologia de estudo de caso

4.1.1 Objetivos

Este estudo de caso teve como objetivo geral obter informações sobre a empresa que possibilitassem a adequação do processo de reuso de requisitos. Este objetivo geral foi decomposto em 2 sub-objetivos:

- Identificar o estado-da-prática em engenharia de requisitos na empresa estudada e
- Identificar o estado-da-prática em reuso na empresa estudada.

4.1.2 Plataforma teórica

A plataforma teórica deste estudo foi apresentada nos capítulos 2 e 3, que apresentam conceitos de Engenharia de Requisitos e de Análise de Domínio. Adicionalmente, a informação contida em alguns artigos foi diretamente contrastada com a realidade da empresa estudada, e é apresentada nas sub-seções 4.2.2.4 e 4.2.3.1. Por último, conhecimentos sobre estudos de caso foram obtidos particularmente em (Flick 2004) e (Martins 2006)

Essa plataforma teórica propiciou o direcionamento do estudo pelo pesquisador, indicando quais informações seria interessante obter da coleta de dados, além de prover o conhecimento necessário para a compreensão das práticas e processos que ocorrem na empresa estudada.

4.1.4 Coleta de dados

Para a realização da coleta de dados foram utilizadas as técnicas Entrevista semi-estruturada, Observação e Análise de conteúdo. Essas atividades foram executadas no decorrer de 6 reuniões presenciais, entre julho de 2007 e janeiro de 2008, com 2 horas de duração cada. Como fonte adicional de informação foram utilizadas as repostas a um questionário e a transcrição de uma entrevista com o diretor de qualidade da empresa estudada, que haviam sido gerados pela pesquisa relatada em (Alves 2007), na qual o autor

também participou. O questionário e o roteiro da entrevista estão disponíveis, respectivamente, em Anexo A - Questionário aplicado na Empresa D em estudo prévio e Anexo B - Roteiro da entrevista realizada na Empresa D em estudo prévio.

Foram realizadas entrevistas individuais com dois analistas de negócios, um analista de testes e um redator de manual, totalizando quatro entrevistados. Na empresa estudada os analistas de negócio são responsáveis pela documentação de requisitos. A técnica de observação foi realizada com os analistas de negócio, durante a realização de duas atividades: documentação de requisitos e análise de requisitos. Durante a análise de conteúdo foram analisados o manual do processo de desenvolvimento de software da empresa, dados armazenados no sistema de controle do processo e diferentes documentos que fazem parte do processo.

Ao término de cada atividade da coleta de dados foram feitas anotações registrando as informações obtidas, ainda na própria empresa. No mesmo dia, as anotações eram passadas para arquivos de texto digital, que compuseram o registro da coleta de dados. Por último, foi escrito um relatório que apresentava textualmente as informações obtidas durante a coleta de dados. Esse relatório foi validado e aprovado pelos analistas de negócio que participaram do estudo e pelo diretor de qualidade da empresa, que confirmaram a correte das informações coletadas.

4.1.5 Análise dos dados

A análise dos dados foi realizada usando a técnica de análise estruturadora do conteúdo, onde o pesquisador identificou todos os fatores que poderiam exercer alguma influência no processo de reuso de requisitos que seria proposto à empresa estudada e agrupou o conteúdo de acordo com esses fatores. O resultado dessa análise foi uma compreensão do contexto atual da empresa, em relação aos objetivos do estudo. Esses resultados são apresentados na seção a seguir.

4.2 Resultados

Nesta seção serão apresentadas as informações sobre a empresa, obtidas durante a realização do estudo de caso. Essas informações foram utilizadas para que o processo de reuso de requisitos fosse adaptado de forma a se adequar à realidade da empresa estudada. Para facilitar a compreensão, as informações foram separadas em três categorias — *Caracterização da Empresa*, que apresenta informações genéricas sobre a empresa; *Processo de Engenharia de Requisitos na Empresa D*, que apresenta informações relacionadas às

atividades de Engenharia de Requisitos que são executadas na empresa e *Práticas de Reuso na Empresa D*, onde são apresentadas informações sobre a situação atual da empresa no que se refere a reuso de *software*.

Por questões de confidencialidade, não será revelado o nome da empresa na qual foi realizado o estudo. Para simplificar a referência à empresa, ela será chamada a partir de agora Empresa D, mantendo seu nome fictício apresentado em (Alves 2007).

4.2.1 Caracterização da Empresa

A Empresa D é especializada no desenvolvimento de sistemas de gestão hospitalar, tendo como clientes hospitais em vários estados do país. Além da sede, localizada em Porto Alegre, a empresa possui filiais em cinco estados e representação em seis estados, sendo que o desenvolvimento de sistemas é realizado na filial de Pernambuco. Com 20 anos de existência e clientes em todas as regiões do país, a maioria dos seus clientes se localiza fora de Pernambuco. De acordo com a classificação feita pelo Banco Nacional de Desenvolvimento Econômico e Social (BNDES 2002), ela se caracteriza como uma empresa de médio porte, por possuir faturamento anual superior a R\$ 10,5 milhões e inferior a R\$ 60 milhões. A Tabela 3 resume algumas das características da Empresa D. Analisando os dados da tabela podemos notar que praticamente metade dos empregados não participa do desenvolvimento de software, e que a equipe de desenvolvimento possui uma baixa experiência.

Tabela 3 - Características da Empresa D

Características	Valores
Idade	20 anos
Faturamento anual	R\$ 40.000.000,00
Domínio de aplicação	Hospitalar
Quantidade de empregados	258
Quantidade de empregados envolvidos com o desenvolvimento de software	130
Experiência média da equipe de desenvolvimento	2 a 4 anos
Clientes	Hospitais
Número de clientes	150
Certificação	ISO 9001 e MPS.Br nível G

A maior parte do trabalho da Empresa D consiste em fazer a manutenção e o desenvolvimento de novas funcionalidades do seu sistema, que atualmente já é considerado um sistema legado. Existe um processo definido para o desenvolvimento de novas funcionalidades, chamado Processo de Desenvolvimento de Software (PDS), aderente ao MPS.Br nível G (SOFTEX 2006).

A Empresa D possui um único grande sistema, cujos módulos podem ser vendidos e implantados isoladamente. Para atender ao mesmo tempo seus aproximados 150 clientes, com necessidades e culturas específicas, o produto é largamente parametrizável. Em 2007, foram implementadas aproximadamente 500.000 linhas de código. Atualmente, o sistema possui um total aproximado de 10.000 formulários e 2.500 tabelas. O sistema existe há 10 anos, e foi desenvolvido em Oracle Developer. Nos próximos anos, existe a previsão de transição do sistema para uma plataforma tecnológica com Orientação a Objetos. Alguns projetos-pilotos vêm sendo realizados nas plataformas Java e Microsoft .NET.

O sistema da Empresa D possui um código-fonte único, ou seja, todos os clientes têm acesso à mesma versão. A periodicidade usual de lançamento de versão é de três meses. *Releases* são liberados em intervalos de uma semana a um mês, contendo principalmente correções de falhas e pequenas implementações. A cada nova versão lançada é enviado aos clientes um documento informando as novidades que foram implementadas. Ocasionalmente, os clientes optam por não atualizar seu *software* para uma versão mais nova. Para diminuir a complexidade que isso pode acarretar, a Empresa D dá suporte apenas às cinco versões mais recentes.

A implantação do sistema em um novo cliente, desde o fechamento do contrato até a conclusão da implantação, dura em média um ano. Durante esse tempo é realizada a configuração do sistema para o hospital, treinamento com os futuros usuários e implementação de novas funcionalidades. A venda é realizada, normalmente, por vendedores em outros estados, que não estão próximos aos desenvolvedores. O distanciamento geográfico, aliado ao desejo de vender, algumas vezes ocasiona distorções entre o esforço de implantação demandado e seus prazos e custos planejados.

Devido à amplitude do sistema e à complexidade do seu domínio, é comum a utilização, em documentos formais e informais, de siglas e termos que não são de conhecimento de todos os funcionários. Não existe um repositório onde os funcionários possam localizar a definição desses termos. Ainda sobre a documentação, pode-se afirmar que toda a documentação gerada, inclusive de requisitos e arquitetura, passa por inspeção.

4.2.1.1 Documentação de ajuda

A equipe de documentação é responsável pela redação da ajuda (*help*) do sistema. Para cada tela do sistema, a ajuda descreve seu objetivo, sua localização dentro do sistema, uma captura da tela, os pré-requisitos para sua utilização, procedimentos passíveis de serem realizados na tela, os resultados gerados pelo seu uso e dicas importantes.

O processo de documentação é definido pelo Processo de Desenvolvimento de Software (PDS). Inicialmente, a equipe de documentação tem uma conversa com o Analista de Negócio responsável pelo novo módulo para ter uma noção geral do que se trata. Em seguida, eles rodam o sistema e simulam o uso incorporando a *persona* de usuário do hospital. Após a redação, a documentação volta para o responsável pelo projeto, que vai realizar a verificação e a validação. De acordo com a equipe de documentação, o ponto crítico nesse processo é a identificação dos pré-requisitos e dos resultados gerados, que são identificados através de análise da tela campo a campo.

Eventualmente o manual fica desatualizado, quando alterações no código não são repassadas à equipe de documentação. Esse problema vem sendo amenizado após a implementação do MPS.Br.

4.2.1.2 Testes

Devido à magnitude do sistema, ao seu alto grau de parametrização, à ausência de mecanismos de rastreabilidade e à não-automação dos testes, atualmente não é possível realizar testes completos do sistema. Dessa forma, versões e *releases* são liberados contendo diversas falhas, o que torna a atualização do sistema nos clientes uma operação arriscada. Dado esse risco, vários clientes optam por não atualizar constantemente o *software*, acarretando na existência de várias versões diferentes do sistema sendo executadas nos clientes. Além do sistema possuir sempre cinco versões sendo mantidas, cada versão do sistema possui diversos *releases*, dificultando ainda mais os trabalhos de testes.

A Empresa D ainda está incipiente em testes, tendo iniciado seus trabalhos específicos nessa disciplina em 2007. Por não haver uma documentação dos requisitos do sistema, os próprios testadores executam o sistema e identificam os requisitos, a partir dos quais se desenvolvem os casos de teste. Dessa forma, os requisitos utilizados para os testes são apenas superficiais, e dependem da interpretação do testador. Espera-se que com uma recuperação dos requisitos do sistema legado esse processo possa ser melhor realizado.

4.2.2 Processo de Engenharia de Requisitos na Empresa D

Como os produtos da Empresa D possuem um código-fonte único, a maioria dos requisitos que são implementados para um cliente são também oferecidos a todos os demais clientes. Essa prática aproveita melhor o esforço de desenvolvimento do que o desenvolvimento de produtos específicos para cada cliente, além de reduzir a complexidade do controle de versões. Mas uma vez que cada cliente possui necessidades específicas,

existem requisitos que se destinam a apenas um ou a um subconjunto de clientes. Por alguns clientes serem concorrentes entre si, é importante que um não possa ter acesso aos requisitos do outro. Deve-se levar em consideração também que diferentes clientes podem possuir diferentes prioridades quanto aos requisitos.

As três maiores fontes de novos requisitos são adequação a legislação e normas, demandas específicas de clientes e criação de novos módulos para agregar ao produto. Quando ocorrem atualizações para se adequar a alguma mudança na legislação, em geral todos os clientes se atualizam para a versão mais nova, o que não necessariamente acontece nos demais casos. As demandas de clientes são levantadas a partir de Pedidos de Atendimento (PDA) e a criação de novos módulos é solicitada pela equipe comercial, salvo raras exceções.

A elicitação de requisitos é feita com uma pequena amostragem dos clientes. Como a base instalada atual de clientes é muito maior que a amostragem, acaba-se incorrendo em uma série de solicitações de mudanças após a implantação de uma nova versão ou módulo do sistema.

O esforço em engenharia de requisitos na Empresa D pode ser dividido em 3 categorias: desenvolvimento de novas funcionalidades, desenvolvimento de novos módulos e recuperação de requisitos do legado.

4.2.2.1 Desenvolvimento de novas funcionalidades

De acordo com o Processo de Desenvolvimento de Software (PDS), o desenvolvimento de novas funcionalidades começa com a criação de um Pedido de Atendimento (PDA), feito remotamente através do site da empresa. O acompanhamento dos PDAs na Empresa D, no decorrer do fluxo do PDS, é realizado através de uma ferramenta desenvolvida internamente.

Os PDAs são divididos em 3 tipos: *Implementação*, quando se solicita mudanças em funcionalidades já existentes; *Novidade*, quando se solicita novas funcionalidades e *Suporte*, quando o cliente está com dúvidas ou deseja avisar da existência de erros no sistema. Internamente, há mais tipos de PDA, mas essa subdivisão fica transparente para o cliente.

Além do tipo, no PDA também consta sua descrição e o caminho da tela à qual está relacionado. Quando são gerados muitos PDAs similares do tipo *Suporte* onde a falha é decorrente de um uso incorreto do sistema pelo cliente, é feita uma análise para saber se o sistema não está excessivamente complicado. Nesses casos, o sistema é modificado para tornar mais fácil o seu uso pelos clientes e, dessa forma, evitar a geração desses PDAs. O PDA não possui uma granularidade bem definida, podendo estar relacionado tanto a pequenas mudanças em uma função quanto ao desenvolvimento de módulos inteiramente novos.

Cada PDA é analisado e priorizado por um Líder de Equipe, passando em seguida para elicitação. A elicitação é necessária porque dificilmente o cliente consegue descrever seu problema de forma clara o suficiente para o correto entendimento pelos desenvolvedores. A elicitação é feita pelo Analista de Negócios com o Cliente Externo, normalmente utilizando-se entrevistas via correio eletrônico e telefone. A partir da elicitação é gerada a Especificação Funcional, que contém uma descrição mais detalhada do pedido do cliente. Esse detalhamento compreende a descrição do que o cliente solicitou, a descrição de como essa solicitação vai ser atendida e uma imagem de como a tela do sistema irá ficar. Além do cliente que originou o PDA, a Especificação Funcional precisa ser validada pelos principais clientes da Empresa D, visto que uma mudança no sistema afeta a todos os clientes.

Em alguns casos a solicitação do cliente é resolvida instruindo-os a utilizar alguma funcionalidade já existente no sistema. Se for constatado que o pedido realmente demanda implementação de software, é gerada a Proposta de Produção e Orçamento (PPO). O PPO contém, além da Especificação Funcional, informações sobre orçamento e prazos do projeto. Esse orçamento é produzido a partir da estimativa de desenvolvimento da funcionalidade em si, além de levar em conta a relevância do cliente e o potencial interesse de outros clientes na funcionalidade em questão. O impacto dessa nova funcionalidade no restante do sistema é analisado apenas superficialmente, ocasionalmente gerando prejuízo quando o esforço de implementação é maior que o previsto inicialmente. Essa falha na análise de impacto é considerada pela Empresa D como um dos grandes problemas do processo vigente. Esse problema poderia ser amenizado através da adoção de rastreamento de requisitos, que não é feito atualmente.

Se o PPO é aprovado dentro do prazo máximo de 30 dias, a Especificação Funcional passa por uma verificação interna e, se aprovada, segue para desenvolvimento.

4.2.2.2 Desenvolvimento de novos módulos

Atualmente a definição das funcionalidades de um novo sistema é feita pelas equipes de *Marketing* e Vendas. As funcionalidades são então detalhadas pelos Analistas de Negócio. Para facilitar o detalhamento das funcionalidades, é feito um levantamento sobre quais das novas funcionalidades já estão implementadas em outros módulos. Esse levantamento é feito manualmente pelo Analista de Negócios, através de execução do sistema.

O processo de desenvolvimento de novos sistemas está sendo alterado, baseando-se no Programa de Melhoria que foi realizado em continuidade ao estudo relatado em (Alves 2007),

do qual este autor também participou. Nesse novo processo se prevê um maior envolvimento do cliente, começando pela definição de metas e escopo.

Ainda no novo processo, caso seja adotado, após a definição de metas e escopo deverá ser realizada uma análise do que está sendo pedido, em conjunto com o gerente de projeto. Se o projeto for considerado viável, o Analista de Negócios irá fazer o trabalho de elicitação com o cliente, compreendendo o problema a ser resolvido e seu contexto. Só então serão elaborados os casos de uso e os requisitos. Os dados contidos na especificação de requisitos serão utilizados como entrada para a elaboração dos casos de teste. Com a especificação pronta, o analista utilizará um repositório central de requisitos para facilitar sua análise, utilizando informações de rastreabilidade e prioridade e identificando conflitos entre os novos requisitos e os já existentes. Como resultado da especificação também serão gerados protótipos da solução.

4.2.2.3 Recuperação de requisitos do legado

Apesar de estar sendo desenvolvido há mais de 10 anos, não existe documentação dos requisitos do sistema legado. As únicas informações registradas sobre o funcionamento do sistema são referentes às mudanças realizadas. A saber: os Pedidos de Atendimento (PDA), a Especificação Funcional e o código fonte.

A recuperação de requisitos do legado é importante para a Empresa D devido à transição que se planeja fazer do sistema para uma plataforma tecnológica cuja linguagem de programação utilize o paradigma de Orientação a Objetos. Em 2006 foi definido internamente um processo para levantar os requisitos do legado, entretanto não se conseguiu colocá-lo em prática. Os fatores apontados pela empresa como responsáveis por essa não adoção foram:

- Alto volume de trabalho da empresa no período de tentativa de adoção;
- Resistência a mudanças pela equipe, especialmente pela dependência gerada entre a empresa e os funcionários pela não-existência dessa documentação;
- Ausência de visualização dos benefícios oriundos desse processo e
- Falta de uma ferramenta que desse suporte ao processo.

Naquele momento se iniciou uma pesquisa por ferramentas disponíveis no mercado para suportar o processo, porém esbarrou-se nos altos custos de aquisição das ferramentas proprietárias. No início de 2007 foi estabelecida uma parceria através da disciplina Especificação de Requisitos e Validação de Sistemas, ministrada no Centro de Informática da Universidade Federal de Pernambuco. Nessa parceria, uma equipe de estudantes da disciplina, incluindo este autor, realizou uma avaliação de ferramentas gratuitas de gerenciamento de

requisitos. O resultado da avaliação está disponível em (Alves 2006 b). Dando continuidade à parceria, este autor apoiou a Empresa D na identificação do *Open Source Requirements Management Tool* (OSRMT) (OSRMT 2008) como a ferramenta gratuita que seria mais apropriada para a mesma. Atualmente, a empresa está avaliando se vai adotar a ferramenta gratuita selecionada, ou uma ferramenta proprietária, para suprir suas necessidades em gerenciamento de requisitos.

Uma parcela do desenvolvimento de novas funcionalidades é realizada por empresas terceirizadas. A dificuldade em se visualizar as interdependências no sistema é considerada a principal causa dos aproximados 1.500 erros novos que são identificados a cada mês no sistema. Esse fator, aliado à perspectiva de transição de plataforma em andamento, são os principais motivadores para a existência de uma documentação dos requisitos do sistema legado.

4.2.2.4 Engenharia de Requisitos Dirigida ao Mercado

Em (Alves 2006 a) foram levantadas na literatura 10 hipóteses sobre a Engenharia de Requisitos Dirigida ao Mercado. Em (Alves 2007) foi dado prosseguimento ao trabalho, relatando observações empíricas sobre as hipóteses. Esse estudo empírico foi realizado com 13 empresas localizadas em Pernambuco, entre elas a Empresa D. Devido aos diferentes contextos dessas empresas, não foi possível em (Alves 2007) realizar afirmações únicas e definitivas sobre as hipóteses. Confrontando especificamente a situação da Empresa D, podemos verificar se as afirmações das hipóteses condizem com a realidade da empresa, como exposto na Tabela 4.

Tabela 4 - Confrontamento da situação da Empresa D com hipóteses sobre Engenharia de Requisitos Dirigida ao Mercado

Hipóteses	Corresponde à realidade da empresa?
1. Tempo de lançamento no mercado é uma preocupação chave em empresas desenvolvedoras de pacotes de software	Sim. Mudanças na legislação, em particular, têm que ser rapidamente refletidas no sistema
2. Os maiores desafios para o crescimento da empresa são relacionados com gerenciamento e marketing, em vez de problemas técnicos	Sim. Em marketing os desafios são basicamente no descompasso entre o que é vendido e o que já existe implementado
3. Requisitos são geralmente inventados pelas empresas desenvolvedoras	Não. Todos os requisitos são levantados junto aos clientes
4. Requisitos são raramente documentados	Parcialmente. Os requisitos do produto não são documentados, mas existe a documentação de todas as mudanças realizadas
5. Seleção de requisitos e planejamento de release são processos cruciais para obter vantagem competitiva	Sim. Há mais demanda do que a empresa é capaz de produzir em um release
6. Relacionamento entre fornecedores e consumidores é geralmente longo, mas com proximidade limitada	Não. O relacionamento é longo e próximo ao cliente
7. A falha no lançamento de produtos é majoritariamente	Não. Fatores culturais têm maior impacto

devido a o produto não satisfazer as necessidades dos clientes	em relação à adoção pelos clientes
8. Requisitos só são validados após o produto ser lançado no mercado	Parcialmente. Os requisitos são validados com alguns clientes antes de ser lançado.
9. Fornecedores de pacotes de software geralmente possuem um processo de engenharia de requisitos ad-hoc	Parcialmente. O processo de ER para mudanças no sistema é definido, documentado e controlado. Para desenvolvimento de novos sistemas o processo é ad-hoc
10. Desenvolvimento dirigido ao mercado apresenta diferenças fundamentais ao processo de ER, em um nível em que práticas tradicionais de ER não podem ser usadas no seu formato original	Não. Práticas de desenvolvimento customizado de software são usadas normalmente.

Um conceito importante na Engenharia de Requisitos Dirigida ao Mercado é o de *Commercial Off-The-Shelf* (COTS), também conhecidos como produto de *software* ou *software* de prateleira (Sommerville 2003). O produto da Empresa D é um sistema de informação caracterizado como *Enterprise Resource Planning* (ERP), tipicamente considerado um COTS. Em (Morisio 2002) é proposta uma definição de COTS dada a partir de dez atributos e seus valores possíveis, definidos pela visão do comprador/usuário de COTS. A Tabela 5 apresenta uma avaliação do sistema desenvolvido na Empresa D de acordo com esses atributos. Comparando os valores que os atributos assumem no produto da Empresa D, com os valores definidos em (Morisio 2002) a partir dos quais um sistema pode ser definido como COTS, comprovamos que o produto da Empresa D pode ser classificado como tal. Mesmo sendo um produto de *software*, a maioria dos hospitais sente a necessidade de contratar a empresa para realizar alterações no código, visando tornar o produto mais adequado às suas necessidades.

Tabela 5 - Avaliação do produto da Empresa D de acordo com a definição de COTS

Atributo	Valores
Origem	Versão especial de comercial
Custo e propriedade	Licenciamento
Modificação requerida	Parametrização
Modificação possível	Customização
Interface	Nenhuma
Empacotamento	Programa executável
Distribuído	Completamente
Tamanho	Enorme
Tipo de Funcionalidade	Vertical
Nível Arquitetural	Núcleo

4.2.3 Práticas de Reuso na Empresa D

A Empresa D está apenas iniciando seus esforços em reuso, portanto ainda não existe nada estabelecido quanto a esse aspecto do desenvolvimento de *software*. Entretanto, foram

identificadas algumas práticas esporádicas e manifestações que podem vir a se tornar as diretrizes de reuso na empresa.

Os produtos da Empresa D são customizáveis de forma a tentar atender todos os clientes com um único código-fonte. Quando vão desenvolver um novo módulo, os analistas de negócio visitam hospitais de diferentes especialidades para elicitare os processos, aos quais esse módulo vai dar suporte, da forma mais genérica possível. Essa preocupação em generalizar os requisitos possui objetivo similar aos objetivos da análise de domínio, mas a análise realizada atualmente se restringe ao escopo do módulo sendo desenvolvido, sem impactar outros módulos que fazem parte do mesmo domínio.

Em relação ao código, a única funcionalidade reusada é o controle de acesso, que é único para todo o sistema. Foi observada a replicação de várias funcionalidades similares, com apenas pequenas variações, que foram desenvolvidas isoladamente. O desenvolvimento isolado dessas funcionalidades acarreta em desperdício de recursos e potencialmente aumenta a quantidade de erros no sistema.

Em projetos recentes, desenvolvidos em Java, observa-se o reuso propiciado pelas características da orientação a objetos, como herança e parametrização (Anastasopoulos 2001). Além disso, estão sendo utilizadas ferramentas de modelagem para geração automática de código, que também é uma forma de reuso. Não ocorre o reuso de modelos entre desenvolvedores, mas ocorre reuso individual e empírico baseado em modelos que um mesmo desenvolvedor já tenha projetado previamente.

A Empresa D está atualmente analisando a viabilidade de utilizar Linguagens de Domínio Específico para aumentar sua produtividade de desenvolvimento, através do reuso por programação generativa (Biggerstaff 1997).

4.2.3.1 Potencial de Reuso na Empresa D

Em (Lucrédio 2007) é apresentada uma pesquisa sobre o sucesso na adoção de reuso de *software* no Brasil. Foram levantados dados de 57 organizações que desenvolvem *software*, a partir dos quais se observou o nível de influência de diferentes fatores sobre a adoção de reuso.

Os fatores de sucesso levantados em (Lucrédio 2007), para uma empresa com mais de 200 empregados, são apresentados na Tabela 6, onde também são apresentados o contexto e a situação da Empresa D em relação a esses fatores. Podemos observar que a empresa se encontra em uma situação desfavorável, mas dado que ela está apenas iniciando seus esforços em reuso isso não chega a ser uma surpresa.

Tabela 6 - Situação da Empresa D em relação aos fatores de sucesso de reuso

Fatores de sucesso	Contexto da empresa	Situação da Empresa
Equipe independente de desenvolvimento de artefatos reusáveis	Não apresenta	Negativa
Família de produtos	Apesar de não desenvolver família de produtos, todos os sistemas estão dentro de um mesmo domínio	Neutra
Abordagem de desenvolvimento de software	Predominantemente paradigma procedural	Negativa
Linguagem de programação	Predominantemente PL-SQL	Negativa
Modelo de qualidade	ISO-9001 e Mps.BR nível G	Positiva
Processo sistemático de reuso	Não apresenta	Negativa
Tipo de artefato reusado	Nenhum	Negativa
Desenvolvimento prévio de artefatos reusáveis	Não apresenta	Negativa

A seguir apresentaremos informações adicionais apresentadas em (Lucrédio 2007), que devem ser levadas em conta para a efetiva adoção de reuso particularmente na Empresa D.

Em empresas com até, aproximadamente, 50 desenvolvedores, é viável a realização de reuso de maneira *ad-hoc*. Quando esse número aumenta, torna-se praticamente impossível fazer reuso sem a utilização de um processo sistemático. Dado que a Empresa D possui mais de 130 empregados diretamente envolvidos com o desenvolvimento de *software*, é notória a importância de um processo bem definido para que ela possa realizar o reuso de software.

Dentre as abordagens de desenvolvimento de *software*, o paradigma procedural, adotado em aproximadamente 95% dos projetos da Empresa D, apresentou a menor taxa de sucesso — 43%. O Paradigma de Orientação a Objetos apresentou 59% de sucesso, ficando atrás de Orientação a Componentes — 64%.

Isoladamente, a experiência da equipe de desenvolvimento não apresentou forte influência sobre o sucesso em reuso. Em conjunto com a abordagem de desenvolvimento, observou-se que o reuso depende fortemente da experiência da equipe quando se utiliza o paradigma procedural. Esse ponto pode ser crítico para a Empresa D caso ela deseje colocar em prática o reuso de código.

O domínio para o qual a empresa desenvolve tem pouca influência no sucesso da adoção de reuso, com exceção do domínio que abrange as ferramentas de desenvolvimento de *software*. Nenhuma menção é feita quanto ao domínio hospitalar, particularmente. Há indícios de que a especialização da organização em um determinado domínio facilite a adoção de reuso.

Existe um movimento de transição dos produtos da Empresa D para a linguagem Java. Essa transição pode facilitar a adoção de reuso na empresa, uma vez que essa linguagem

influencia três fatores de sucesso em reuso: *abordagem de desenvolvimento de software*, *linguagem de programação* e *experiência da equipe*. Ademais, os projetos que já foram implementados em Java utilizaram ferramentas do tipo *Computer-Aided Software Engineering* (CASE), que segundo (Lucrédio 2007) é um fator de forte influência sobre a adoção de reuso.

As empresas que reusam apenas código apresentaram um sucesso menor do que as empresas que reutilizaram demais artefatos de desenvolvimento. Em particular, empresas onde as especificações de requisitos eram reusadas obtiveram 70% de sucesso, enquanto a taxa de sucesso das empresas onde só se reutilizava o código foi 56%. Isso é um indício de que o reuso de requisitos tende a facilitar o reuso nas demais etapas do processo de desenvolvimento de *software*.

4.3 Considerações finais

No início deste capítulo foi apresentada a metodologia empregada para a condução do estudo de caso, dando inicialmente uma visão geral e em seguida a aprofundando.

Em seguida, foram apresentados os resultados obtidos no estudo de caso. Para facilitar a compreensão, os resultados foram separados em três categorias: *Caracterização da Empresa*, *Processo de Engenharia de Requisitos na Empresa D* e *Práticas de Reuso na Empresa D*.

No próximo capítulo será apresentado o processo proposto a partir das observações apresentadas neste capítulo e da plataforma teórica apresentada nos capítulos 2 e 3. O capítulo 5 constitui-se na principal contribuição do trabalho aqui relatado.



5 Processo adaptado de reuso de requisitos

Em trabalhos prévios realizados com a Empresa D (Alves 2007) (Alves 2006 b) (Pereira 2007), foi identificado um forte interesse pela adoção de reuso de requisitos. O interesse da empresa coincidiu com o interesse dos pesquisadores em adquirir mais conhecimento sobre esse tipo de reuso, de uma forma que fosse enriquecedora tanto para empresa, que teria acesso facilitado a esse conhecimento; quanto para a academia, que ampliaria seu conhecimento sobre o estado da prática. Nesses termos foi estabelecida uma parceria entre ambas as partes, cujo resultado está sendo apresentado no presente documento.

No decorrer deste capítulo será apresentado o processo de reuso de requisitos que foi estabelecido especificamente para a Empresa D. Esse processo foi baseado na metodologia *Feature-Oriented Domain Analysis* (FODA) (Kang 1990). A adaptação da metodologia, em vez da sua utilização precisamente como descrito na literatura, é importante para que o processo atenda ao contexto específico onde será inserido, como retratado especificamente com FODA em (Vici 1998) (Zalman 1996), e mais genericamente em (Morisio 2000).

O referido processo foi validado em reunião com o analista de negócios responsável por essa parceria na empresa. Nessa reunião foram sugeridas pequenas alterações no processo, que já estão refletidas no processo apresentado nas próximas seções.

Para evitar a notória ambigüidade que o acrônimo da metodologia utilizada possui na língua portuguesa, o processo adaptado será nomeado Análise de Domínio Orientada a Características adaptado para a Empresa D (ADOC-D). Esse processo, apesar de poder ser adotado por qualquer organização, foi definido especificamente para o contexto da Empresa D, portanto antes de sua adoção é necessária uma análise criteriosa sobre a adequação do mesmo à nova realidade à qual ele venha a ser inserido.

Os principais fatores que levaram à decisão da abordagem de análise de domínio para a realização de reuso, e em particular da metodologia FODA, foram a existência de um domínio único no qual a empresa é especializada, no caso o domínio hospitalar e para o qual são feitos vários produtos complementares, e a característica apresentada por FODA de trabalhar não só com a especificação mas também com a arquitetura, possibilitando o reuso em outras etapas do ciclo de desenvolvimento de *software*. A seguir é apresentado o meta-modelo utilizado para descrever o processo, e posteriormente o processo propriamente dito.

5.1 Software Process Engineering Metamodel - SPEM

O SPEM (OMG 2005) é um meta-modelo (Seidewitz 2003) utilizado para especificar processos de desenvolvimento de *software*. Ele é uma extensão da *Unified Modeling Language* - UML (OMG 2007 a)(OMG 2007 b), definido pela mesma organização que especifica o UML, o *Object Management Group* (OMG). Para se modelar e representar visualmente um processo em SPEM são utilizados os mesmos diagramas presentes em UML, como diagramas de classe, de casos de uso e de atividade, fazendo uso de regras e estereótipos específicos. Como os desenvolvedores, em geral, já possuem familiaridade com UML, o SPEM tende a ser facilmente compreendido por eles.

Os estereótipos utilizados para a modelagem do ADOC-D foram:

- *Process* (Processo) – o processo propriamente dito, ou um processo externo com o qual ele interage;
- *Phase* (Fase) – descreve um trabalho do processo que possui critérios de início e término bem definidos;
- *Discipline* (Disciplina) – é um tipo de pacote que agrupa atividades dentro de um processo de acordo com um tema em comum. É uma especialização de *ProcessPackage*;
- *ProcessRole* (Papel) – representa os realizadores do processo, atribuindo responsabilidades sobre trabalhos específicos;
- *Document* (Documento) – é um artefato que contém informações produzidas ou utilizadas pelo processo. Pode ser tanto formal quanto informal, como correspondências eletrônicas ou diagramas.
- *Activity* (Atividade) – descreve um trabalho que é realizado ou auxiliado por um papel. Possui entradas e saídas;
- *Guidance* (Orientação) – é um elemento que provê informações mais detalhadas sobre algum componente do processo. Exemplos de *Guidance* são diretrizes, técnicas, métricas, exemplos, listas de verificação e modelos (*templates*);

A Figura 10 apresenta a notação gráfica dos principais estereótipos de SPEM. Para elaborar os diagramas SPEM foi utilizada a ferramenta Microsoft Visio 2007 (Microsoft 2007), com o perfil de SPEM para Microsoft Visio (FIPA). Exemplos de utilização do SPEM estão disponíveis em (Araújo 2006) e (Nardini 2008).

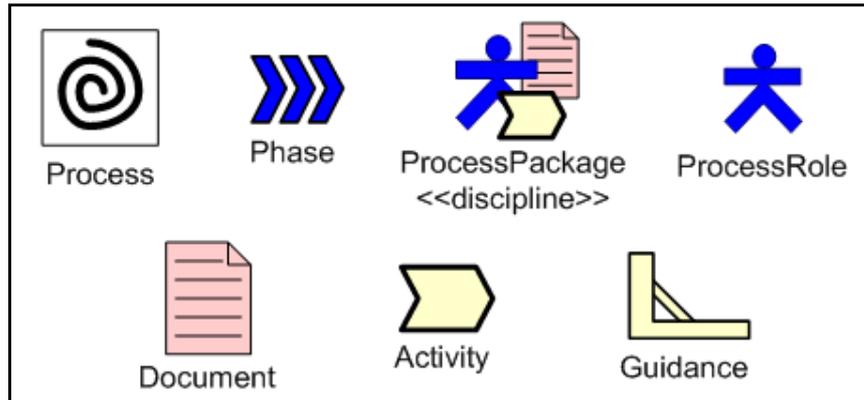


Figura 10 – Estereótipos do SPEM utilizados em ADOC-D

Nas próximas seções o processo ADOC-D será detalhado, provendo as informações necessárias para sua execução. Nos apêndices estão disponíveis modelos de documentos utilizados no processo, doravante denominados *templates* para evitar confusão com os modelos (de contexto e de domínio) gerados durante a execução do processo.

5.2 Processo - Fases

O processo ADOC-D foi separado em duas fases: Engenharia de Domínio e Engenharia de Requisitos. Cada fase recebe e gera novas informações para a outra, caracterizando uma retro-alimentação, conforme representado na Figura 11.

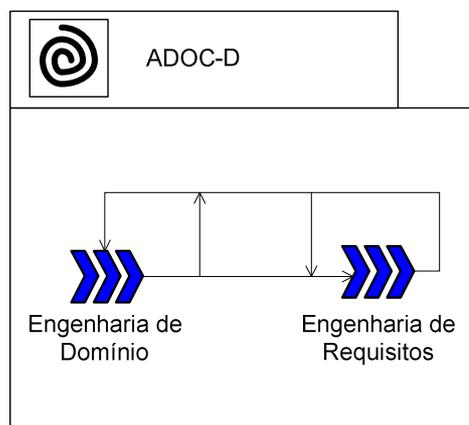


Figura 11 – Fases do processo ADOC-D

5.3.1 Engenharia de Domínio

A Engenharia de Domínio compreende o desenvolvimento para reuso, e deve ser executada antes da primeira execução da fase de Engenharia de Requisitos. Nela são adquiridos conhecimentos sobre o domínio e gerados ou atualizados os seus modelos. Essa fase compreende duas disciplinas – Análise de contexto e Modelagem de domínio – ilustradas pela Figura 12. As disciplinas serão detalhadas na seção 5.4.

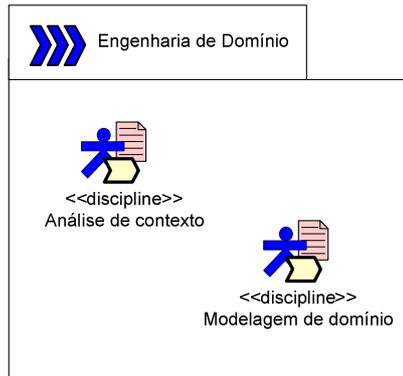


Figura 12 – Disciplinas da fase Engenharia de Domínio

Quando essa fase é executada visando a modelagem de um novo domínio, seu início ocorre partir da solicitação de modelagem de um novo domínio por um Líder de Equipe. Se o domínio a ser trabalhado já foi modelado, essa fase é iniciada pela solicitação de revisão do domínio, também pelo líder de equipe. O término da fase acontece com a validação do modelo de domínio que foi gerado ou atualizado, pelo líder de equipe.

5.3.2 Engenharia de Requisitos

A Engenharia de Requisitos compreende o desenvolvimento com reuso. Sua execução só deve ocorrer após a fase Engenharia de Domínio ter sido realizada ao menos uma vez. A partir do conhecimento adquirido na Engenharia de Domínio, nesta fase se define ou se atualiza os requisitos de um produto que faça parte do domínio já modelado. A fase de Engenharia de Requisitos possui uma única disciplina – Requisitos, ilustrada na Figura 13.

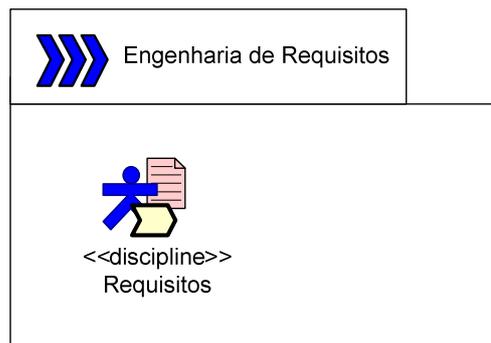


Figura 13 – Disciplina da fase Engenharia de Requisitos

Essa fase pode ser executada tanto para o desenvolvimento de um novo produto quanto para a realização de mudanças em um produto já existente, a depender da solicitação que lhe deu início. O término é caracterizado pela documentação ou atualização da documentação dos requisitos do referido produto.

5.3 Processo - Papéis

Os papéis são uma abstração dos realizadores dos trabalhos, independente da pessoa que o executa ou do cargo que ocupa. Um mesmo papel pode ser realizado por diferentes cargos ou pessoas, assim como diferentes papéis podem ser desempenhados pelos mesmos cargos ou pessoas. Os papéis que fazem parte do processo ADOC-D são apresentados na Figura 14.



Figura 14 – Papéis do processo ADOC-D

5.4.1 Especialista do Domínio

Este papel representa uma pessoa que tenha profundo conhecimento sobre o domínio. Ele atua como uma ponte de informações entre os clientes e a empresa, repassando seu conhecimento sobre o domínio para os analistas.

A Figura 15 apresenta as responsabilidades do Especialista do Domínio, sob a forma de diagrama de casos de uso. A interação com as atividades, em SPEM, é classificada em 2 estereótipos:

- *<<perform>>* - utilizado quando o papel representado realiza efetivamente a atividade, sendo o responsável por ela;
- *<<assist>>* - indica que o papel representado apóia a realização da atividade, mas não é o responsável pela sua execução.

As atividades serão detalhadas nas seções que falam sobre as suas respectivas disciplinas.

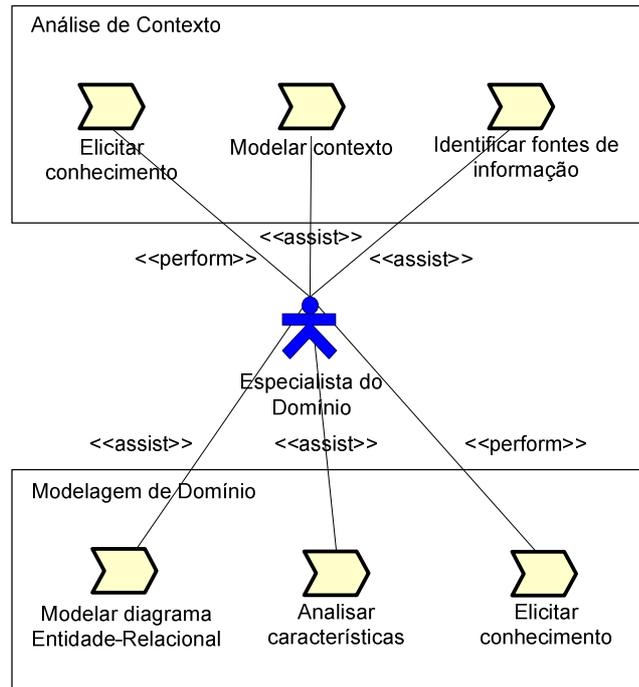


Figura 15 – Diagrama de casos de uso do papel Especialista do Domínio

5.4.2 Analista de Domínio

Este é o principal papel do processo ADOC-D. Representa a pessoa que possui profundo conhecimento sobre a metodologia e, preferencialmente, boas noções sobre o domínio em que irá atuar. As atividades das quais ele participa são apresentadas na Figura 16.

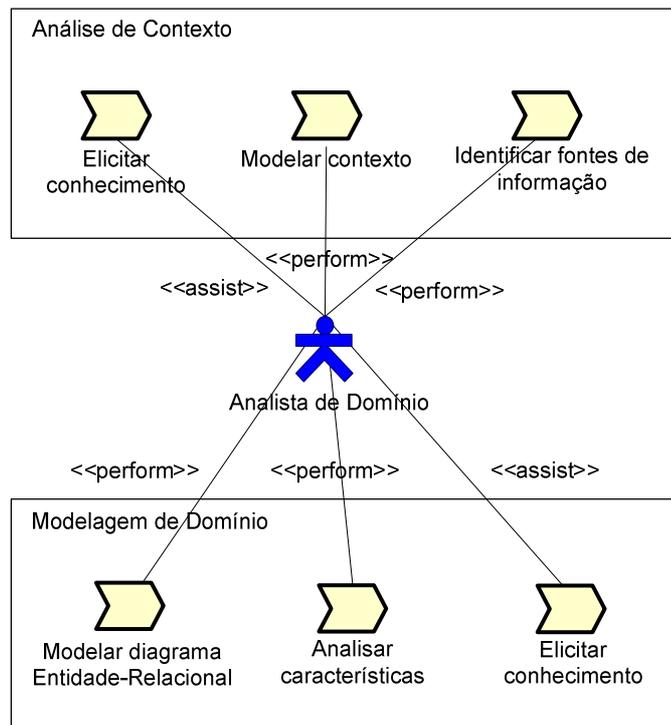


Figura 16 – Diagrama de casos de uso do papel Analista de Domínio

5.4.3 Líder de Equipe

O papel Líder de Equipe representa a pessoa que é responsável por conduzir a fase Engenharia de Domínio. É mais importante que ele tenha habilidades gerenciais e conhecimento sobre a empresa do que conhecimento sobre o domínio. As atividades nas quais esse papel participa estão apresentadas na Figura 17.

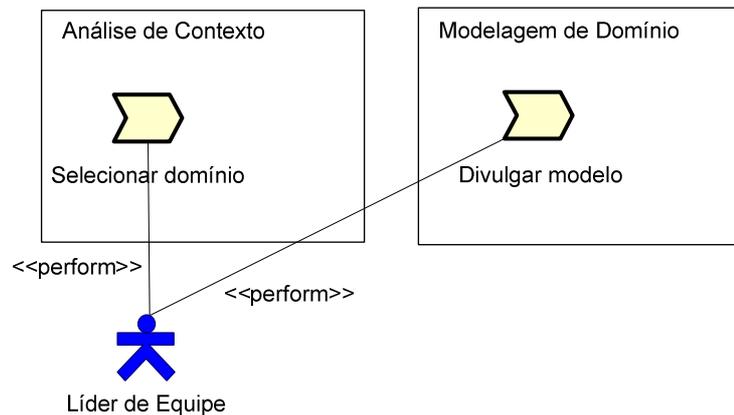


Figura 17 – Diagrama de casos de uso do papel Líder de Equipe

5.4 Processo - Disciplinas

As fases do ADOC-D são organizadas em disciplinas, para facilitar seu controle e compreensão. Ao todo são três disciplinas, a saber: Análise de Contexto, Análise de Domínio e Engenharia de Requisitos. A seguir serão apresentadas cada uma das disciplinas do processo ADOC-D.

5.5.1 Análise de Contexto

Esta disciplina tem como objetivo definir o escopo do domínio a ser explorado. Para tanto é necessário analisar relacionamentos entre o domínio e elementos externos. Durante a execução dessa disciplina é gerado o Modelo de Contexto do domínio. O diagrama de pacotes da disciplina Análise de Contexto, ilustrado na Figura 18, apresenta os elementos que a compõem.

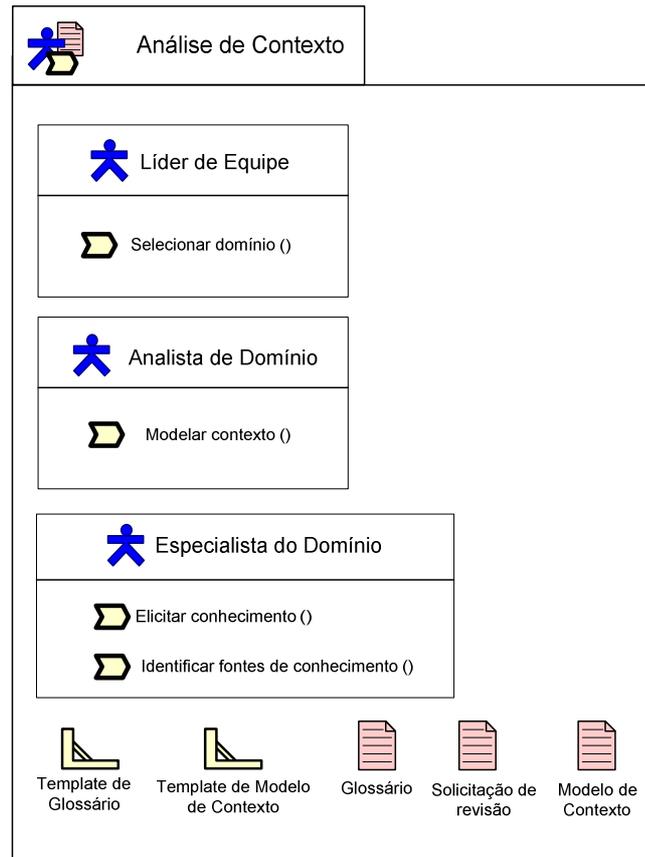


Figura 18 – Diagrama de pacotes da disciplina Análise de Contexto

A Figura 19 exibe o diagrama de atividades dessa disciplina. Inicialmente o Líder de Equipe seleciona o domínio que será analisado, levando em conta informações de demanda comercial e de estratégia da empresa. O domínio selecionado pode ser um domínio já modelado pela empresa, cujo Modelo de Contexto tenha alguma solicitação de revisão. Nesse caso o especialista de domínio irá elicitar conhecimento sobre o domínio para poder identificar se a revisão sugerida é pertinente, ou se há alguma outra alteração no domínio que precisa ser refletida com uma alteração no Modelo de Contexto. Como o domínio hospitalar já se encontra maduro, as chances de mudanças desse tipo ocorrerem são pequenas.

Se o domínio ainda não houver sido modelado pela empresa, o Líder de Equipe redigirá uma breve descrição do mesmo, iniciando a construção do documento chamado Modelo de Contexto. Além disso, o Líder de Equipe deve criar o glossário, que será atualizado assincronamente à medida que novas definições são identificadas. Em seguida o Analista de Domínio, apoiado pelo Especialista do Domínio, irá fazer a modelagem do contexto propriamente dita. Assume-se que o Especialista do Domínio possui conhecimento suficiente sobre o domínio para a realização dessa atividade, mas em qualquer momento que seja necessário ele pode elicitar mais conhecimento sobre o domínio com as fontes de informação.

Na modelagem de contexto o analista irá prosseguir com a elaboração do Modelo de Contexto, gerando o diagrama de fluxo de dados do domínio conforme exposto no capítulo 3. O diagrama de estrutura, utilizado juntamente ao diagrama de fluxo de dados na metodologia FODA, não é usado no processo ADOC-D. Esse diagrama foi abolido para diminuir o esforço necessário nessa fase do processo e por se considerar que o impacto negativo de sua ausência é pequeno, uma vez que os principais componentes do diagrama de estrutura também estão presentes no diagrama de fluxo de dados.

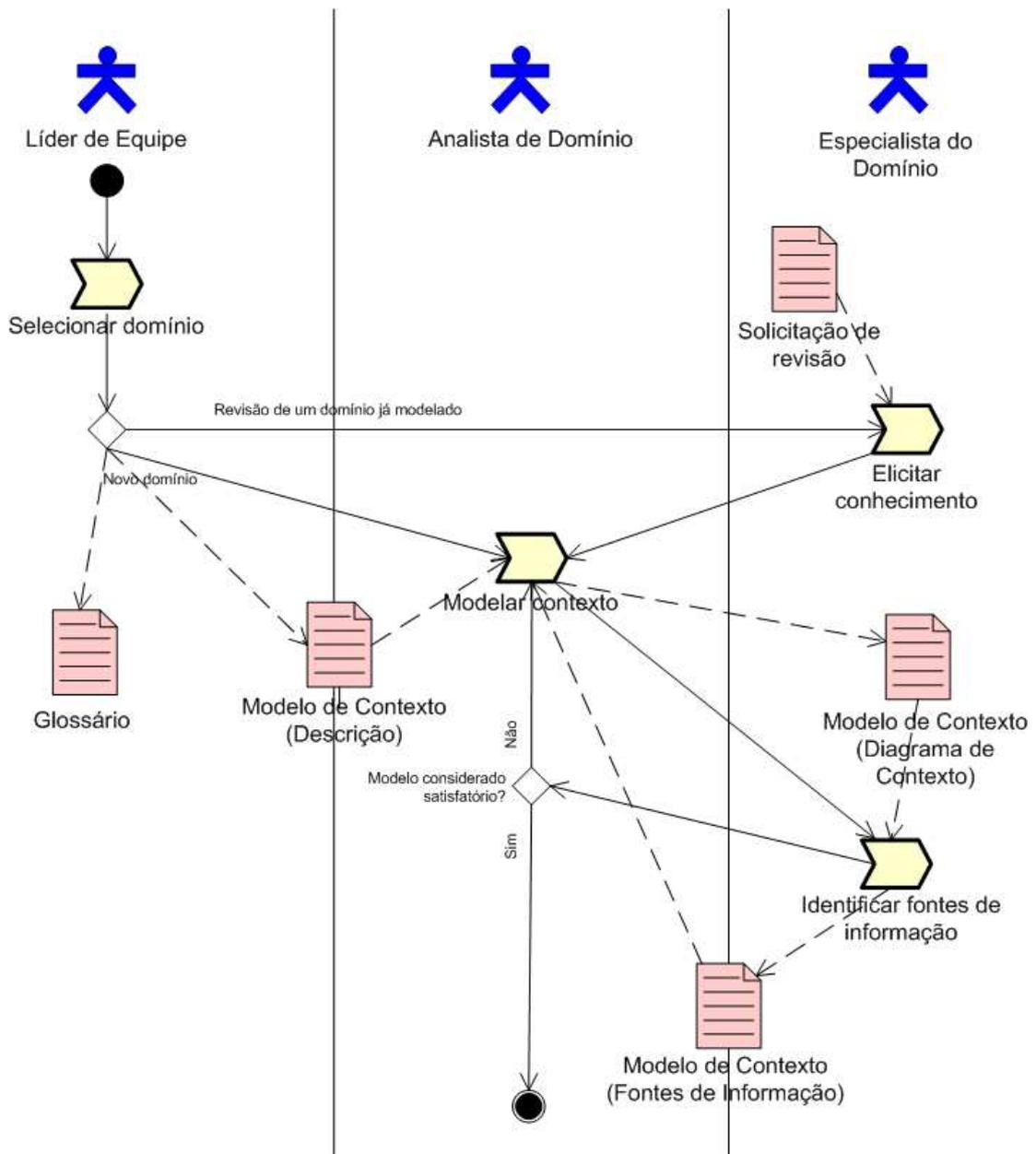


Figura 19 – Diagrama de atividades da disciplina Análise de Contexto

Após uma modelagem inicial do contexto, o Especialista do Domínio identifica as fontes de informação que podem ajudar na compreensão do domínio. Essas fontes serão utilizadas tanto para a elicitação de conhecimento sobre o contexto do domínio quanto para a posterior modelagem do domínio. As fontes podem ser de qualquer origem, desde artigos em revista e livros até clientes da empresa, incluindo documentos presentes no hospital, como prontuários e receitas médicas. Nesse momento o Líder de Equipe apóia a atividade interagindo com os clientes identificados para avaliar suas disponibilidades e interesses em apoiar a análise do domínio, além de estabelecer uma parceria que possibilitará a elicitação de conhecimentos com os referidos clientes.

Enquanto o Modelo de Contexto não é considerado satisfatório e acordado pelos 3 papéis envolvidos na disciplina, o processo fica em ciclo entre as atividades Modelar contexto e Identificar fontes de informação. Quando o modelo é enfim considerado satisfatório, as atividades da disciplina são encerradas e as atividades da disciplina Modelagem de domínio são iniciadas, utilizando-se o modelo então gerado.

Eventualmente, durante as atividades da disciplina Análise de Contexto, serão identificados termos e acrônimos cujo significado não sejam entendíveis a leigos no domínio. É importante que nesses casos ocorra a atualização do glossário, com a inclusão de um novo item ou a alteração de um item já registrado.

5.5.2 Modelagem de Domínio

O objetivo desta disciplina é representar o conhecimento do domínio em um modelo que possibilite a compreensão do mesmo e que diminua o esforço necessário para a definição de requisitos de produtos dentro desse domínio. Esta é a disciplina mais dispendiosa do processo, tanto em recursos humanos quanto em tempo. O resultado da execução dessa disciplina é a criação ou a atualização de um Modelo de Domínio. Os componentes dessa disciplina estão apresentados na Figura 20.

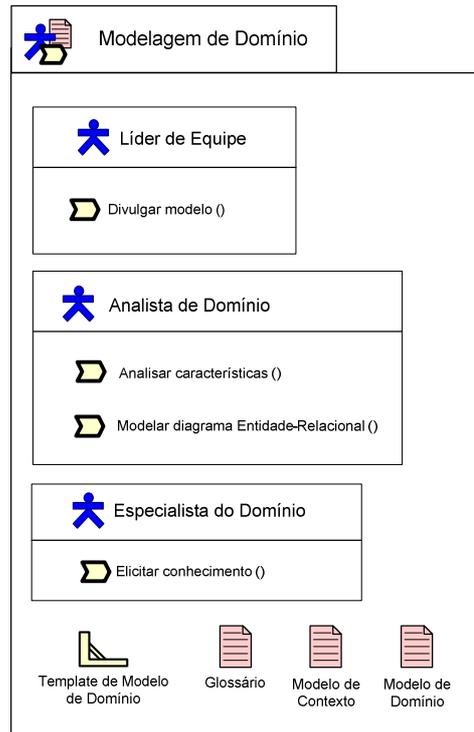


Figura 20 – Diagrama de pacotes da disciplina Modelagem de Domínio

O diagrama de atividades da disciplina Modelagem de Domínio é apresentado na Figura 21. A disciplina começa com a elicitação de conhecimento sobre o domínio, realizada pelo Especialista do Domínio. Para tanto ele recebe o Modelo de Contexto e o Glossário. Um dado particularmente importante para a realização dessa atividade, contida no Modelo de Contexto, são as fontes de informação.

Da mesma forma que na disciplina Análise de Contexto, na disciplina Modelagem de Domínio pode ocorrer a atualização do glossário assincronamente, à medida que novos termos ou acrônimos forem identificados ou alterados.

Uma vez que o Especialista do Domínio tiver adquirido conhecimento o suficiente, ele apóia o Analista de Domínio na realização da análise de características, conforme apresentado no capítulo 3. Juntamente às descrições das características, o processo ADOC-D expande a modelagem de características da metodologia FODA com um novo elemento: ligação com requisitos. A definição dessas ligações não precisa ocorrer já na primeira vez em que o domínio for modelado, uma vez que possivelmente não existe um sistema cujos requisitos se relacionam com o domínio.

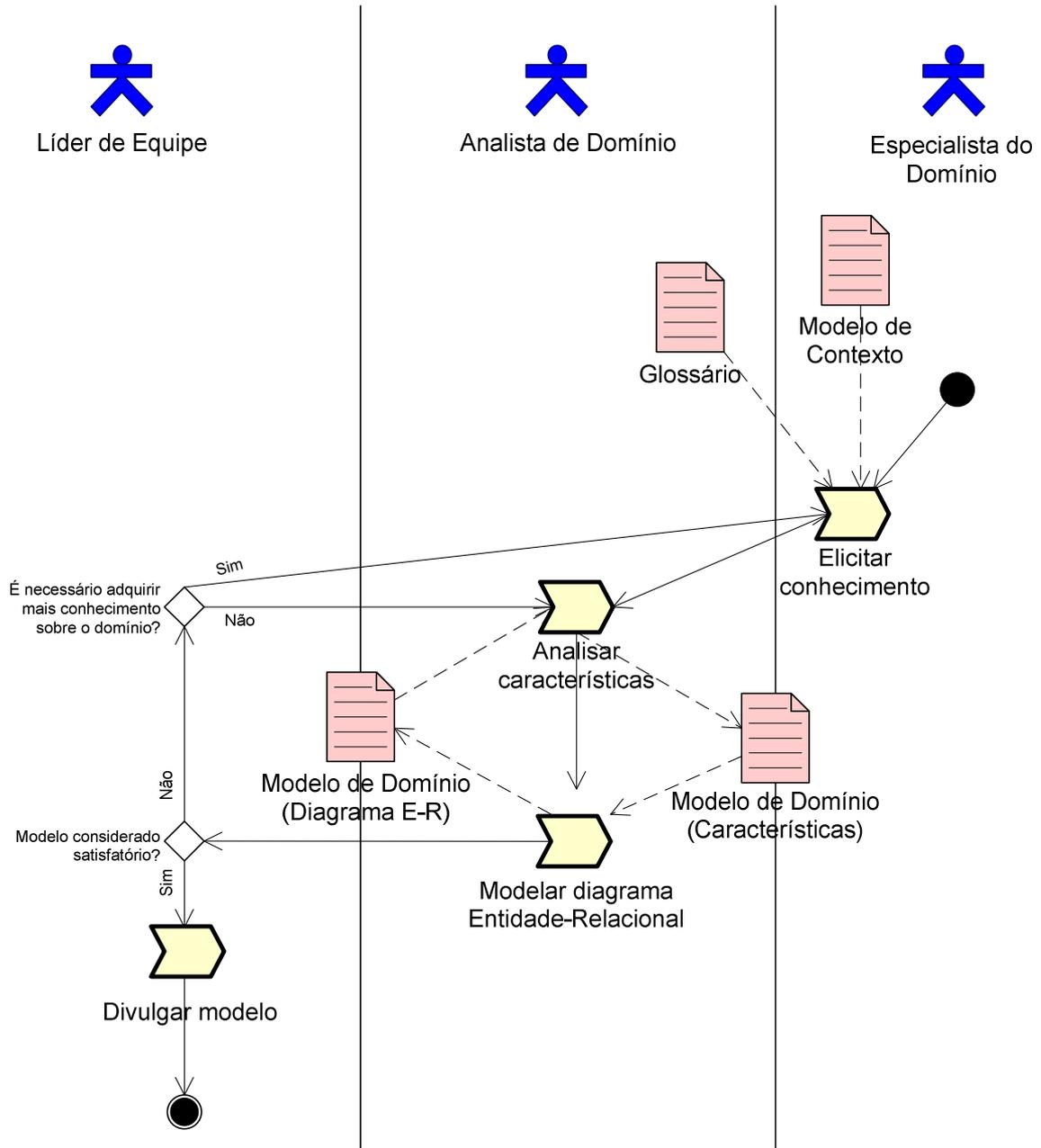


Figura 21 – Diagrama de atividades da disciplina Modelagem de Domínio

Com a ligação entre uma característica e os requisitos aos quais ela está relacionada, será mais fácil para o Analista de Requisitos, posteriormente, identificar oportunidades de reuso. Além disso, quando eventuais mudanças acontecerem no domínio será mais fácil fazer os produtos se adequarem a essas mudanças devido à rastreabilidade gerada por esses relacionamentos.

Com o modelo de características gerado ou atualizado, dentro do Modelo de Domínio, o Analista de Domínio constrói o Diagrama Entidade-Relacional (E-R), apoiado pelo Especialista do Domínio. Essa atividade também é descrita na metodologia FODA, apresentada no capítulo 3. Enquanto os 3 papéis envolvidos na disciplina não considerarem

que o Modelo de Domínio, a execução volta para a atividade Analisar características ou a atividade Elicitar conhecimento, a depender da necessidade de conhecimento no momento.

Quando o Modelo de Domínio é considerado satisfatório, cabe ao Líder de Equipe realizar a divulgação do modelo dentro da empresa. É importante que o modelo gerado seja acessível a todos os funcionários que tenham alguma interação com o domínio, senão ele não terá utilidade. É salutar, ainda, que essas pessoas saibam como solicitar revisão do modelo e como acessar o glossário.

5.5.3 Requisitos

A Empresa D possui um processo definido de Engenharia de Requisitos (ER) para mudanças no sistema, e está definindo um processo de Engenharia de Requisitos que dê suporte ao desenvolvimento de novos sistemas. Como o foco deste trabalho é em reuso de requisitos, e não em requisitos propriamente dito, julgou-se pertinente definir a disciplina Requisitos a partir da interação com o processo da empresa, expandindo-o com o que for necessário para a realização de reuso. Dessa forma, a disciplina ficou definida como ilustrado na Figura 22, contendo o processo de Engenharia de Requisitos da empresa e as atividades Análise do Modelo de Domínio e Solicitação de revisão de domínio.

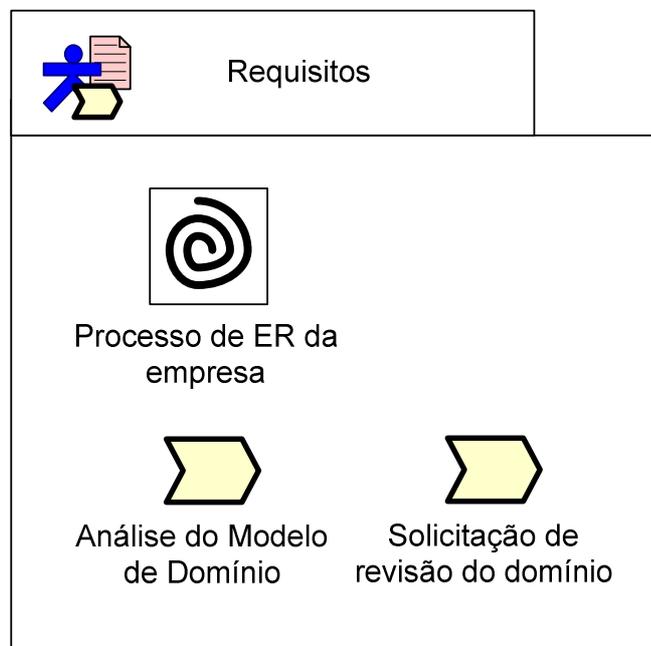


Figura 22 – Diagrama de pacote da disciplina Requisitos

Durante a Análise do Modelo de Domínio, o analista responsável pelo processo de Engenharia de Requisitos utiliza o Modelo de Domínio para compreender o domínio no qual está elicitando os requisitos de um produto. Quando estiver realizando a documentação dos requisitos, o analista verifica no Modelo de Domínio quais elementos do domínio farão parte

do produto, estabelecendo o relacionamento entre as características e os requisitos. Caso seja observada alguma inconsistência ou lacuna no domínio, o analista deve solicitar a revisão do domínio, informando qual problema foi encontrado. A revisão do domínio é realizada na fase Engenharia de Domínio.

5.5 Considerações finais

No início deste capítulo foi introduzido o processo de reuso de requisitos adaptado para a Empresa D, denominado Análise de Domínio Orientada a Características adaptado para a Empresa D (ADOC-D).

Na seção 5.1 foi apresentado o meta-modelo *Software Process Engineering Metamodel* (SPEM), utilizado para descrever o processo ADOC-D. Nas seções seguintes foi apresentado cada componente do processo ADOC-D – suas fases, seus papéis e suas disciplinas.

No próximo capítulo são apresentadas as principais conclusões do trabalho, indicando também trabalhos futuros que podem vir a ser realizados a partir deste.



6 Conclusões

Neste trabalho foi apresentado um processo de reuso de requisitos para uma empresa em específico, denominado Análise de Domínio Orientada a Características adaptado para a Empresa D (ADOC-D). Esse processo usa a abordagem de Análise de Domínio, tendo como base a metodologia *Feature-Oriented Domain Analysis* (FODA). Este trabalho constitui-se uma contribuição para a ampliação do conhecimento sobre o assunto nos países de língua portuguesa, facilitando o acesso a informações sobre essa metodologia.

O estudo de caso relatado neste trabalho é um retrato verídico do estado-da-prática em uma empresa desenvolvedora de *software* brasileira. Por esse motivo, é um elemento de valor para pesquisas que visam compreender o cenário nacional de desenvolvimento de *software*, especialmente sobre requisitos. Esse relato também pode ser utilizado para identificar lacunas que podem ser preenchidas por mais pesquisa em Engenharia de *Software*.

O processo ADOC-D foi definido para uma empresa desenvolvedora de *software* em específico, não sendo aconselhável a sua adoção integral em outras empresas. Entretanto, ele é um ponto de partida para que as empresas façam uma análise crítica do mesmo e o adaptem para o seu contexto. Em particular, para as empresas cujo ambiente se assemelhe ao apresentado no estudo de caso, esse trabalho deve demandar menos esforço que a criação de um processo de reuso diretamente a partir de FODA.

Pelo seu caráter de transferência tecnológica, uma das principais contribuições deste trabalho é o processo propriamente dito. O ADOC-D foi validado com a empresa para o qual foi definido, mas devido à magnitude do esforço que ele demanda para implantação, e o longo tempo de retorno, é improvável que essa implantação ocorra no curto prazo. Eventualmente, à medida que a empresa amadureça seu conhecimento sobre o tema, novos elementos serão incorporados ao processo. Vale ressaltar o crescimento do interesse estratégico da empresa na área de Engenharia de Requisitos, a partir das pesquisas realizadas junto a ela. Adotando processos e práticas presentes na literatura, acreditamos que as empresas possam melhorar seus processos e, com isso, aumentar a satisfação dos seus clientes.

6.1 Trabalhos futuros

O *Software Process Engineering Metamodel* (SPEM) foi utilizado para descrever o processo ADOC-D. Durante a pesquisa para compreender esse meta-modelo, foi observada a existência de pouca orientação sobre o uso do mesmo. Basicamente, o que existe é a

especificação oficial e alguns trabalhos científicos que utilizam o meta-modelo para descrever um processo, servindo como exemplos de uso. A elaboração de um manual de modelagem com SPEM é um trabalho futuro que pode trazer benefícios para toda a comunidade de Engenharia de *Software*.

Devido a restrições de escopo, o presente trabalho delimitou sua Análise de Domínio às fases referentes a requisitos. Um possível trabalho futuro é a ampliação do processo ADOC-D para compreender também a fase de Modelagem de Arquitetura.

Uma lacuna do processo aqui definido é a ausência de métricas que avaliem não só sua implantação quanto ao retorno sobre o investimento por ela gerada. A definição dessas métricas é um trabalho futuro que faz parte dos planos do autor.

Existem poucos estudos que avaliam o impacto da adoção de reuso em uma empresa de *software*. Em particular, o impacto que novas práticas de reuso podem trazer para estratégias de negócio e decisões tecnológicas futuras. O acompanhamento da implantação do ADOC-D na Empresa D é um trabalho futuro que pode enriquecer a literatura sobre reuso de *software*, e sobre reuso de requisitos em particular, não só pela avaliação do resultado em si, mas também pelas lições que podem ser aprendidas nesse processo.

Um ponto que pode dificultar a adoção do ADOC-D é a ausência de uma ferramenta que lhe dê suporte. Em particular, uma ferramenta que facilite a modelagem e a manutenção da árvore de características é um componente que deve ser melhor pesquisado, uma vez que poucas ferramentas existentes hoje podem ser utilizadas com esse fim. Possivelmente, existe uma demanda pelo desenvolvimento de um *software* dessa natureza.



Referências

- (Almeida 2007) ALMEIDA, E.S.; ALVARO, A.; GARCIA, V. C.; MASCENA, J. C. C. P.; BURÉGIO, V. A. A.; NASCIMENTO, L. M.; LUCRÉDIO, D; MEIRA, S. R. L. **C.R.U.I.S.E: Component Reuse in Software Engineering**. C.E.S.A.R e-book, Brazil, 2007.
- (Alves 2006 a) ALVES, C. ; Pereira, S. ; CASTRO, J. B. . **A Study in Market Driven Requirements Engineering**. In: Workshop on Requirements Engineering, 2006, Rio de Janeiro. Workshop on Requirements Engineering, 2006.
- (Alves 2006 b) ALVES, C. ; PEREIRA, S. ; ANDRADE, R. ; PIMENTEL, J. ; SANTOS, G. . **Um Estudo Empírico sobre Práticas de Engenharia de Requisitos junto a Empresas de Pacotes de Software**. 2006.
- (Alves 2007) ALVES, C. ; PEREIRA, S. ; SANTOS, G. ; PIMENTEL, J. ; ANDRADE, R. . **Preliminary Results from an Empirical Study in Market-Driven Requirements Engineering**. In: Workshop em Engenharia de Requisitos, 2007, Toronto. Anais do WER07 - Workshop em Engenharia de Requisitos, 2007.
- (Anastasopoulos 2001) ANASTASOPOULOS, M.; GACEK, C. (2001). **Implementing Product Line Variabilities**. In: Proceedings of the 2001 Symposium on Software Reusability (SSR'01), Recife: Federal University of Pernambuco, 2001.
- (Arango 1994) ARANGO, G. **Domain Analysis Methods**. Software Reusability. Chichester, England: Ellis Horwood, 1994.
- (Araújo 2006) ARAÚJO, A. R. S. **Agile Game Process – Metodologia Ágil para Projetos de Advergames**. Universidade Federal de Pernambuco, 2006. Trabalho de Graduação (Graduação em Ciências da Computação).
- (Ahmad 2004) AHMAD, F; AZIZ, U. **A survey of domain analysis techniques and domain reuse in Pakistan**. NUCES-FAST, Lahore, Pakistan. In: Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International, 24-26 dezembro de 2004. p.434- 439.
- (Basili 1991) BASILI, V. R.; ROMBACH, H. D. **Support for comprehensive reuse**. In IEE Software. Engineering Journal, 6(5):303–316, setembro de 1991.
- (Biggerstaff 1997) BIGGERSTAFF, T. J. **A Perspective of Generative Reuse**. In:

- Annals of Software Engineering, 1997
- (BNDES 2002) BNDES. **Carta Circular nº 64**. Rio de Janeiro, 14 de outubro de 2002.
- (Chen 1976) CHEN, P.P. **The Entity-Relationship Model: Toward a Unified View of Data**. ACM Trans. on Database Systems, Vol.1, No.1, março de 1976, pp. 1-36.
- (Cohen 1992) COHEN, Sholom G., et al. **Application of Feature-Oriented Domain Analysis to the Army Movement Control Domain**. Software Engineering Institute. Carnegie Mellon University. Pittsburgh, Pennsylvania, 1992.
- (Cornwell 1996) CORNWELL, P.C. **HP domain analysis: Producing useful models for reusable software**. Hewlett-Packard. Journal, article 5 de agosto de 1996.
- (Creps 1992) CREPS, R. E.; SIMOS, M. A. et al. **The STARS conceptual framework for reuse processes, software technology for adaptable, reliable systems (STARS)**. Technical Report, DARPA: novembro de 1992.
- (Czarnecki 2000) CZARNECKI, K.; EISENECKER, U. **Generative Programming - Methods, Tools, and Applications**. Addison-Wesley, Boston, MA, 2000.
- (Ferré 1999) FERRÉ, X.; VEGAS, S. **An Evaluation of Domain Analysis Methods**. In: 4th CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'99). Heidelberg (Germany): 14-15 de junho de 1999.
- (FIPA) (FIPA) **SPEM profile for Microsoft Visio**. In: <http://www.pa.icar.cnr.it/cossentino/FIPAmeth/metamodel.htm>. Acesso em 17 de dezembro de 2007.
- (Flick 2004) FLICK, U. **Uma Introdução à Pesquisa Qualitativa - Segunda Edição**. Bookman Editora: Porto Alegre, 2004.
- (Freeman 1987) FREEMAN, P. **A Conceptual Analysis of the Draco Approach to Constructing Software systems**. IEEE Trans. on Software Engineering, SE-13(7):830-844, julho de 1987.
- (Gartner 1995) GARTNER GROUP. **A Guide for Estimating Client/Server Costs**. 1995.
- (Holibaugh 1992) HOLIBAUGH, R. **Joint integrated avionics working group (JIAWG) object-oriented domain analysis method (JODA)**. Version 3.1. Special Report CMU/SEI-92-SR-3. Software

- Engineering Institute. Carnegie Mellon University, Pittsburgh, PA: novembro de 1993.
- (Kang 1990) KANG, K., et al. **Feature-Oriented Domain Analysis (FODA) Feasibility Study (CMU/SEI-90-TR-21, ADA 235785)**. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1990.
- (Keepence 1995) KEEPENCE, M. M.; SMITH, S. **SMARTRe Requirements: Writing Reusable Requirements**. Proceedings of IEEE Symposium on Engineering of Computer Based Systems. Tucson, Arizona, USA, março de 1995, ISBN 07803-2531-1.
- (Kotonya 1997) KOTONYA, G.; SOMMERVILLE, I., **Requirements Engineering: Processes and Techniques**, ISBN 0471972. John Wiley, 1998.
- (Lam 1998 a) LAM, W. **A Case-Study of Requirements Reuse Through Product Families**. In Annals of Software Engineering, vol. 5, pp. 253-277. 1998.
- (Lam 1998 b) LAM, W.; JONES, S.; BRITTON, C. **Technology Transfer for Reuse: A Management Model and Process Improvement Framework**. ICRE, 1998.
- (Loucopoulus 1995) LOUCOPOULUS, P.; KARAKOSTAS, V. **System Requirements Engineering**. Mc Graw-Hill: London, 1995.
- (Lubars 1989) LUBARS, M.D. **The Idea Design Environment**. Microelectronics and Computer Technology Corp.: Software Engineering, 1989.
- (Lucrédio 2007) LUCRÉDIO, D. ; BRITO, K. S. ; ALVARO, A. ; GARCIA, V. C. ; ALMEIDA, E. S. ; FORTES, R. P. M. ; MEIRA, S. R. **Software Reuse: The Brazilian Industry Scenario**. Journal of Systems and Software (in press), 2007.
- (Maccario 1997) MACCARIO, P.M. **The domain analysis integrated in an object oriented development methodology**. In: Proceedings of the Eighth Workshop on Institutionalizing Software Reuse. Columbus, Ohio, março de 1997.
- (Martins 2006) MARTINS, G. de A. **Estudo de caso: uma estratégia de pesquisa**. São Paulo: Atlas, 2006.
- (Microsoft 2007) **Microsoft Office Online - Visio Homepage**. In: <http://office.microsoft.com/pt-br/visio/>. Acesso em 12 de janeiro de 2008.

- (Morisio 2000) MORISIO, M. **Diversity in Reuse Diversity in Reuse Processes** . University of Maryland: julho/agosto de 2000.
- (Morisio 2002) MORISIO M., TORCHIANO, M., **Definition and classification of COTS: a proposal**. In: Int. Conference on COTS Based Software Systems. ICCBSS 2002, Orlando, FL, fevereiro de 2002.
- (Nardini 2008) NARDINI, E. **SPEM on test: the SODA case study**. Alma Mater Studiorum –. Universit`a di Bologna. SAC'08. Fortaleza, Ceará, 16-20 de março de 2008.
- (Neighbors 1980) NEIGHBORS, J. **Software Construction Using Components**. Ph.D. Thesis, ICS-TR-160, University of California at Irvine, 1980.
- (Nuseibeh 2000) NUSEIBEH, B. & Garlan, D. **Requirements Engineering: A Roadmap**. Steve Easterbrook, 2000.
- (OMG 2005) OMG Middleware Specifications. **Software Process Engineering Metamodel**, v1.1. janeiro de 2005.
- (OMG 2007 a) OMG Unified Modeling Language (OMG UML) Infrastructure, V2.1.2. In: <http://www.omg.org/spec/UML/2.1.2/>. Acesso em 18 de janeiro de 2008
- (OMG 2007 b) OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2. In: <http://www.omg.org/spec/UML/2.1.2/>. Acesso em 18 de janeiro de 2008
- (OSRMT 2008) Open Source Requirements Management Tool. In: <http://www.osrmt.com>. Acesso em 9 de janeiro de 2008.
- (Parnas 1976) PARNAS, D. L. **On the Design and Development of Program Families**. IEEE Trans. Software Eng. 2(1): 1-9 (1976)
- (Pereira 2007) PEREIRA, S. C. **Um Estudo Empírico sobre Engenharia de Requisitos em Empresas de Software**. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Pernambuco, 2007.
- (Pinheiro 2003) PINHEIRO, F.A.C.; LEITE J.C.S.P.; CASTRO J.F.B. **Requirements Engineering Technology Transfer: An Experience Report**. Journal of Technology Transfer, 28, 159–165, 2003.
- (Prieto-Diaz 1990) PRIETO-DIAZ, R. **Domain Analysis: An Introduction**. Software Engineering Notes 15, 2 (Abril de 1990): 47-54.
- (Rajat 2003) RAJAT, S. R.; ARTHUR, J. D. (2003) **Requirements**

- Management Tools: A Quantitative Assessment.** Technical Report TR-03-10, Computer Science, Virginia Tech.
- (Robertson 2000) ROBERTSON, S. **Identifying and involving the stakeholders.** The Atlantic Systems Guild Ltd, 2000.
- (Robertson 2006) ROBERTSON, J.; ROBERTSON, S. **Mastering the Requirements Process.** Addison-Wesley, London, 1999.
- (SEI 2007) SEI 2007. **Domain Engineering and Domain Analysis.** In: <http://www.sei.cmu.edu/str/descriptions/deda.html>. Acesso em 20 de novembro de 2007.
- (Seidewitz 2003) SEIDEWITZ, E. **What Models Mean.** IEEE Software 20(5):26-32, setembro/outubro de 2003.
- (Simos 1995) SIMOS, M.A. **Organization domain modeling (ODM): formalizing the core domain modeling life cycle.** In: Mansur Samadzadeh and Mansour Zand, editors, Proceedings of the Symposium on Software Reusability. SSR '95, pages 196-205, Seattle, Washington, 28-30 de abril de 1995.
- (SOFTEX 2006) SOFTEX. **MPS.BR – Melhoria de Processo do Software Brasileiro.** 2º Seminário Interno do Observatório Digital SOFTEX. Campinas, 13 de fevereiro de 2006.
- (Sommerville 2003) SOMMERVILLE, I. **Engenharia de Software.** Sexta edição. Pearson / Prentice Hall: Brasil, 2003.
- (Simos 1995) SIMOS, M. A. **Organization Domain Modeling (ODM) Guidebook.** Version 1.0. Unisys STARS Technical Report STARS-VC-A023/011/00, Advanced Projects Research Agency. Março, 1995.
- (Thayer 1997) THAYER, R & DORFMEN, M. **Software Requirements Engineering - Second Edition.** IEEE Computer Society Press, Los Alamitos, California, 1997.
- (Vici 1998) VICI, A.D.; ARGENTIERI, N.; MANSOUR, A.; D'ALESSANDRO, M. ; FAVARO, J. **FODAcOm: An Experience with Domain Analysis in the Italian Telecom Industry.** In: Fifth International Conference on Software Reuse - ICSR'98, p. 166.
- (Villegas 2001) VIELLEGAS, O. L.; LAGUNA, M. **Requirements Reuse for Software Development.** RE 01 Doctoral Workshop. 5th IEEE International Symposium on Requirements Engineering. Toronto, Canadá, p 27-31, agosto de 2001
- (Zalman 1996) ZALMAN, N. S. **Making the Method Fit: An Industrial**

Experience in Adopting Feature-Oriented Domain Analysis (FODA). In: Proceedings of the 4th International Conference on Software Reuse table of contents, IEEE Computer Society Washington, DC, USA, 1996.

(Zave 1997)

ZAVE, P. **Classification of Research Efforts in Requirements Engineering.** ACM Computing Surveys, vol. 29, no. 4, 1997.



Apêndice A – *Template* de Glossário

Glossário

Definições dos principais termos e acrônimos

Domínio: **<Nome do domínio>**

Data de origem

<Data de criação do documento>

Responsável

<Nome da pessoa responsável pelo documento>

Índice

<É importante a existência de um índice no glossário, assim como a ordenação alfabética dos termos, para facilitar a localização dos termos no documento. Os editores de texto atuais permitem a manutenção do índice e da ordem alfabética automaticamente>

<Termo 1 – página X

Termo 2 – página X...>

Política de atualização

<É interessante a descrição de uma política de atualização do glossário nesta seção. A política serve para evitar que atualizações sejam feitas sem critério, desvirtuando o propósito do glossário>

Definições

<Nesta seção se encontra o glossário propriamente dito, com a definição dos principais termos e acrônimos deste domínio>

<Termo 1>

- <Definição>. <Exemplos, referências...>

<Termo 2>

- <Definição>. <Exemplos, referências...>



Apêndice B – *Template* de Modelo de Contexto

Modelo de Contexto

Domínio <Nome do domínio>

Data de origem	<Data de criação do documento>
Responsável	<Nome da pessoa responsável pelo documento>
Versão do Documento	<Versão do Documento> - <Descrição da última alteração>

Descrição do domínio

<Nesta seção é apresentada uma breve descrição do domínio a ser analisado. Deve ser detalhado o suficiente apenas para que o Analista de Domínio saiba qual o domínio que deve ser analisado >

Fluxo de dados

<Nesta seção é apresentado o fluxo de dados do domínio. Para cada diagrama de fluxo de dados pode haver comentários textuais que ajudem na sua compreensão >

Fluxo de Dados <Número do fluxo de dados>





<Aqui fica a descrição textual de algum aspecto do diagrama que se considere pertinente esclarecer melhor>

Fontes de informação

<Nesta seção são registradas as fontes de informação que podem ser consultadas para se adquirir conhecimento sobre o domínio. Caso haja muitas fontes de um tipo em específico, além de clientes, pode-se criar uma outra subseção para abrangê-los>

Clientes

- <Nome do hospital> - <Informações de contato>

Outras fontes

- <Nome da fonte, autor, onde encontrar...>



Apêndice C – *Template* de Modelo de Contexto

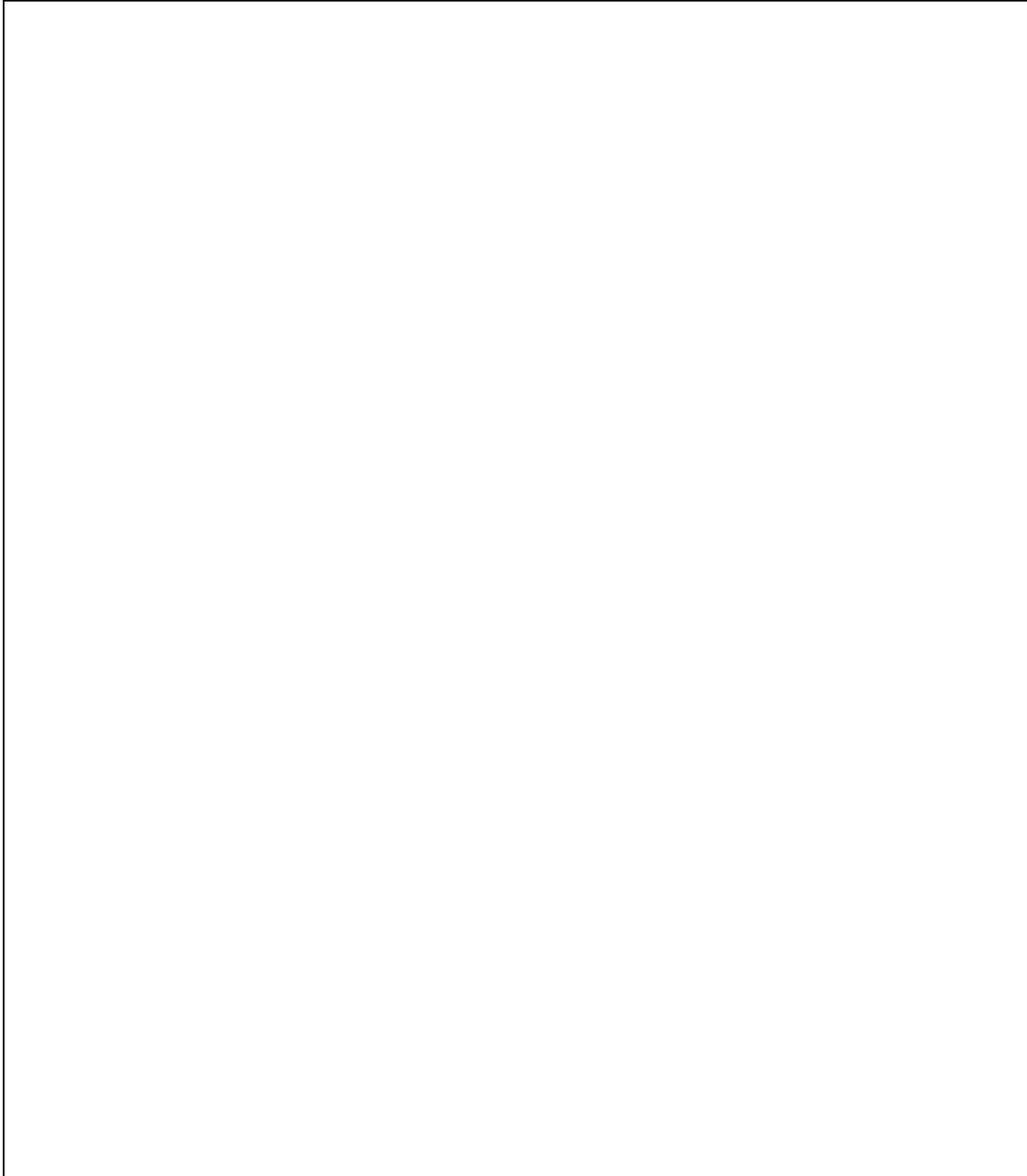
Modelo de Domínio

Domínio: **<Nome do domínio>**

Data de origem	<Data de criação do documento>
Responsável	<Nome da pessoa responsável pelo documento>
Versão do Documento	<Versão do Documento> - <Descrição da última alteração>

Árvore de características

<Nesta seção é apresentada a árvore de características do domínio. Caso a árvore não caiba em uma página, pode ser dividida em várias páginas>



Características

<Nesta seção se encontra a descrição das características e seus relacionamentos com requisitos>

<Característica 1>

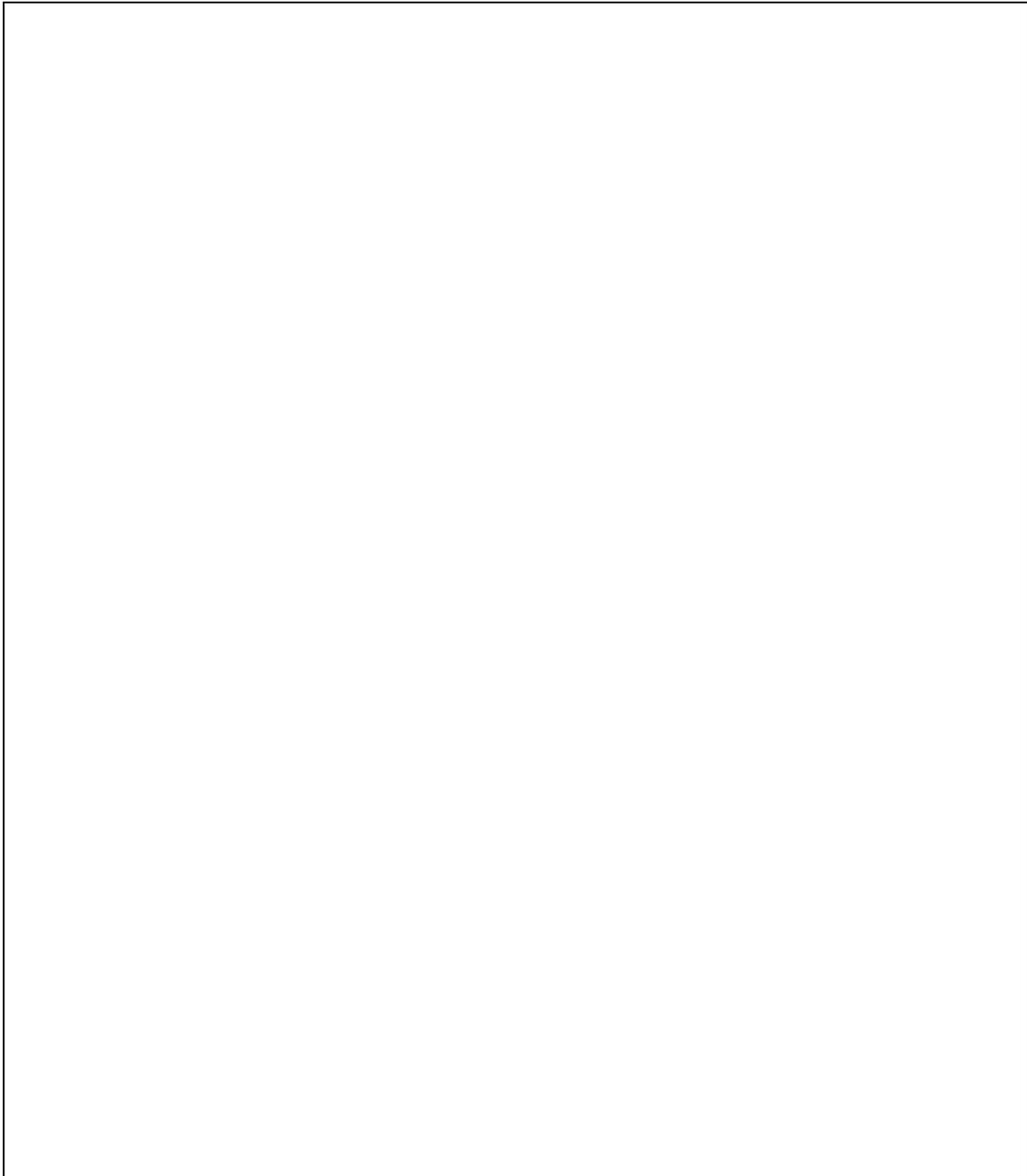
- <Descrição>
- <Requisitos relacionados>

<Característica 2>

- <Definição>. <Exemplos, referências...>

Diagrama Entidade-Relacional

<Nesta seção se encontra o diagrama entidade-relacional do domínio>



Anexo A - Questionário aplicado na Empresa D em estudo prévio

Questionário referente à caracterização de empresas desenvolvedoras de produtos de software

Objetivos	O propósito deste questionário é obter informações sobre empresas desenvolvedoras de produtos de software em Pernambuco.
Descrição da pesquisa	Este estudo faz parte do projeto de pesquisa financiado pelo CNPq <i>Um Framework de Engenharia de Requisitos para o Desenvolvimento de Produtos de Software</i> e coordenado pela Dra. Carina Alves. A meta da pesquisa é difundir conhecimento de ponta gerado na área de engenharia de software para sua efetiva utilização por empresas de software brasileiras. Essa pesquisa objetiva realizar um estudo sobre práticas de engenharia de software adotadas por empresas desenvolvedoras de produtos de software. Em particular, a pesquisa foca no processo de engenharia de requisitos e aspectos mercadológicos do desenvolvimento de produtos de software. Na primeira fase da pesquisa, realizamos um amplo estudo da literatura nessas áreas. Na segunda fase, que se inicia com esse questionário, iremos realizar um estudo empírico com empresas de software de Pernambuco. O objetivo do estudo empírico é investigar práticas, oportunidades e desafios encontrados durante o desenvolvimento e comercialização de produtos de software. Um dos principais resultados dessa pesquisa é a identificação, desenvolvimento e difusão de melhores práticas em engenharia de software para empresas de software.
Benefícios	As empresas participantes desse estudo terão acesso a todos os relatórios e artigos publicados sobre a pesquisa. As empresas receberão um relatório com os resultados do estudo contendo <i>guidelines</i> sobre como melhorar o processo de engenharia de requisitos durante o desenvolvimento de produtos de software, assim como aumentar a satisfação de clientes. Outro importante benefício desse projeto de pesquisa é estimular o debate entre profissionais da indústria de software.
Confidencialidade	Toda informação disponibilizada pelas empresas participantes do estudo é confidencial. Toda publicação feita desse estudo apresentará somente resultados agregados das informações obtidas. Nenhuma informação específica de cada empresa será disponibilizada.
Maiores informações	Qualquer dúvida ou esclarecimento sobre esse questionário ou sobre o projeto de pesquisa, entrar em contato com Carina Alves no seguinte e-mail: carinafrota@gmail.com

Glossário

As seguintes definições são adotadas nesse questionário:

Cliente – Organização ou pessoa compradora de um sistema de software fornecido por uma empresa desenvolvedora de software.

Engenharia de Requisitos - Atividade que constitui a definição, análise e validação dos requisitos, funcionalidades e restrições do sistema de software a ser desenvolvido.

Engenharia de software – Disciplina que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais da especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação.

Família de produtos - conjunto de sistemas que usam software intensivamente, compartilhando um conjunto de características comuns e gerenciadas, que satisfazem as necessidades de um segmento particular de mercado.

Produto de software – sistema de software desenvolvido por uma empresa e disponível para diversos clientes num mercado consumidor.

Requisitos do sistema - Uma necessidade do usuário, característica, funcionalidade ou atributo que deve ser alcançado por um sistema para satisfazer um contrato, um padrão, uma especificação ou outros documentos.

Release planning – planejamento feito por uma empresa desenvolvedora de produto de software para lançar uma nova versão do produto no mercado.

Time to market – Tempo necessário desde a idealização e desenvolvimento de um produto de software até a sua comercialização no mercado.

1. Dados de Identificação

Nome da empresa: _____

Nome do Respondente: _____

Cargo: _____

E-mail: _____

Data da Resposta (dd/mm/aaaa): ____ / ____ / ____

2. Caracterização da empresa

1. Quantos anos têm a empresa?

_____ ano(s).

2. Qual é o faturamento anual da empresa?

_____ Reais.

3. Qual o número (total) de funcionários da empresa?

_____ funcionário(s).

4. Quantos funcionários estão envolvidos diretamente com atividades de desenvolvimento de software?

_____ funcionário(s).

5. Qual é a experiência média da equipe de desenvolvimento de sua empresa?

- a. 1 ano ou menos
- b. 2 a 4 anos
- c. 5 a 10 anos
- d. mais de 10 anos
- e. Não sei

6. Qual(is) são o(s) domínio(s) de aplicação(ões) dos produtos desenvolvidos por sua empresa?

- | | |
|---|--|
| a. <input type="checkbox"/> Administrativo | j. <input type="checkbox"/> Farmacêutico |
| b. <input type="checkbox"/> Comunicação | l. <input type="checkbox"/> Químico e Petroquímico |
| c. <input type="checkbox"/> Alimentos e Bebidas | m. <input type="checkbox"/> Simulação |
| d. <input type="checkbox"/> Comércio (atacado e varejo) | n. <input type="checkbox"/> Telecomunicações |
| e. <input type="checkbox"/> Controle de Tráfego | o. <input type="checkbox"/> Desenvolvimento de Ferramentas |
| f. <input type="checkbox"/> Educacional | p. <input type="checkbox"/> Ambientes de Desenvolvimento de Software |
| g. <input type="checkbox"/> Financeiro | p. <input type="checkbox"/> Outro, por favor
especifique_____ |
| h. <input type="checkbox"/> Higiene e Limpeza | q. <input type="checkbox"/> Não sei |
| i. <input type="checkbox"/> Entretenimento | |

7. A empresa possui algum modelo ou certificação de qualidade de software?

- a. Sim
b. Não
c. Não sei

8. Caso a resposta à questão 7 seja afirmativa, qual modelo ou certificado é utilizado?

- a. ISO 9001/9000-3
b. CMMI
c. Outro. Por favor,
especifique_____
- d. Não sei

3. Informações sobre produtos de software

9. Qual(is) tipo(s) de produtos de software são desenvolvido(s) pela empresa?

- a. Embarcado
b. Jogos
c. Sistemas para Web
d. Sistemas de Informação
e. Sistemas Inteligentes
f. Sistemas de Tempo Real
g. Outro. Por favor,
especifique_____
- h. Não sei

10. Quantos produtos de software são desenvolvidos pela empresa?

_____ produto(s).

11. Quais são os produtos?

Instrução para as questões 12 - 36

Se sua empresa desenvolve mais de um produto de software, por favor, responder as questões a seguir para cada produto de software oferecido pela empresa.

12. Qual é a plataforma do produto de software?

- a. Desktop
- b. Telefone celular
- c. Palm
- d. Outra. Por favor, especifique:

e. Não sei

13. Que tipo de produto de software sua empresa está desenvolvendo?

- a. Produto entregue sem customização para vários clientes
- b. Produto customizado para vários clientes
- c. Produto para uso interno da sua empresa
- d. Componente subcontratado de um sistema de software
- e. Outro. Por favor, especifique:

f. Não sei

4. Informações sobre o processo de engenharia de software

19. Qual abordagem de desenvolvimento de software é utilizada?

- a. Projeto estruturado
- b. Orientada a Objetos
- c. Orientada a Componentes
- d. Orientada a Aspectos
- e. Outra. Por favor, especifique:

f. Não sei

20. Qual(is) a(s) linguagem(ns) de programação são utilizada(s) no desenvolvimento do produto?

- a. Pascal
 - b. Object Pascal (Delphi)
 - c. C
 - d. C++
 - e. ASP
 - f. PHP
 - g. Visual Basic
 - h. Java
 - i. .NET (C#, ASP.NET, VB.NET)
 - j. Outra, por favor, especifique
-

21. Qual(is) papel(is) você exerce no projeto de desenvolvimento do produto? (Selecione mais de uma alternativa se necessário)

- a. Gerente de Projeto
 - b. Gerente de Marketing
 - c. Analista de Sistemas
 - d. Engenheiro de Requisitos
 - e. Arquiteto de Software
 - f. SQA (*Software Quality Assurer*)
 - g. Líder de Desenvolvimento
 - h. Programador
 - i. Gerente de Testes
 - j. Testador
 - l. Representante de clientes/usuários
 - m. Gerente de configuração
 - n. Designer
 - o. Outro, por favor, especifique
-

22. O produto de software em desenvolvimento faz reuso de partes/componentes?

- a. Sim
- b. Não
- c. Não sei

23. Quais são os três principais desafios enfrentados pela empresa durante o processo de desenvolvimento do produto?

24. Existe um processo de engenharia de requisitos bem definido?

- a. Sim
- b. Não
- c. Não sei

25. Explique como é realizado o processo de engenharia de requisitos.

26. O(s) cliente(s) participa(m) do processo de definição dos requisitos para um novo produto de software?

- a. Sim
- b. Não
- c. Não sei

27. Caso a resposta à questão 26 seja afirmativa, explique como é a participação do cliente.

28. Os requisitos do sistema são documentados?

- a. Sim
- b. Não
- c. Não sei

29. Caso a resposta à questão 28 seja afirmativa, explique como os requisitos são documentados.

30. Qual a importância dos seguintes fatores durante o desenvolvimento do produto?

	Irrelevante (1)	Neutro (2)	Importante (3)	Muito importante (4)	Não sei ()
Time to market	(1)	(2)	(3)	(4)	()
Release planning	(1)	(2)	(3)	(4)	()
Qualidade do produto	(1)	(2)	(3)	(4)	()
Satisfação do cliente	(1)	(2)	(3)	(4)	()
Resultados de vendas	(1)	(2)	(3)	(4)	()

5. Informações sobre os clientes do produto

31. Quantos clientes já compraram o produto?

_____ cliente(s).

32. Quem são os principais clientes do produto?

33. Quais são os três principais benefícios oferecidos pelo produto para atrair clientes?

34. Quais são as três principais dificuldades enfrentadas pela empresa durante a venda do produto?

35. A sua empresa mantém um programa de relacionamento com o cliente (CRM)?

- a. Sim
- b. Não
- c. Não sei

36. Caso a resposta à questão 35 seja afirmativa, descreva o programa de relacionamento com o cliente.



Anexo B - Roteiro da entrevista realizada na Empresa D em estudo prévio

Metodologia:

Realizar entrevistas semi-estruturadas com empresas estudadas. Tempo médio de duração: 1 hora.

Objetivos:

1. Entender como as empresas realizam o processo de engenharia de requisitos,
2. Investigar formas de interação e envolvimento com clientes durante o processo de desenvolvimento de pacotes de software,
3. Descobrir os principais desafios enfrentados durante o processo de requisitos.

Perguntas:

A Empresa

1. Qual a sua posição na empresa?
2. Comente quais são as suas atribuições e atividades diárias.
3. Fale sobre os produtos de software desenvolvidos por sua empresa.
4. Como é conduzido o marketing da empresa e dos produtos? Existe uma estratégia bem definida?
5. Explique como é feito o contato com novos clientes.
6. Como é a interação com clientes já conquistados?
7. A interação é considerada satisfatória?
8. Quais mecanismos a sua empresa usa para avaliar a satisfação dos clientes?
9. Explique como os produtos são desenvolvidos (tipo de processo, atividades realizadas, técnicas e ferramentas utilizadas).

O Processo de Requisitos

10. O que são requisitos para sua empresa?
11. Explique como é realizado o processo de engenharia de requisitos.
12. Quem são os responsáveis pela análise de requisitos na sua empresa?
13. Quem define os requisitos do sistema?
14. Os clientes são envolvidos na decisão para escolher quais novas funcionalidades serão oferecidas nas futuras versões do sistema?
15. Qual é a frequência em que novas versões do produto são disponibilizadas para os clientes?
16. A sua empresa permite que um cliente requisiute customizações específicas do produto?
17. Se a resposta anterior for afirmativa, como é realizado o controle das várias versões?
18. A sua empresa usa alguma ferramenta CASE para auxiliar o processo de requisitos?
19. Como os requisitos são documentados (ex. linguagem natural, casos de uso, ferramentas)?
20. Quais informações são documentadas sobre os requisitos?
21. Como os requisitos são priorizados e selecionados para serem implementados numa versão do sistema?
22. Quais são os principais fatores para ter um processo de engenharia de requisitos bem sucedido?
23. Como a sua empresa poderia melhorar o processo de requisitos?