

VICTOR OLIVEIRA HERMSDORF

**MODELAGEM DA ATIVIDADE DE ELICITAÇÃO DO PROCESSO DE
ENGENHARIA DE REQUISITOS: UMA ABORDAGEM UTILIZANDO DINÂMICA
DE SISTEMAS**

Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do Programa de
Pós-Graduação em Ciência da
Computação, para obtenção do título de
Magister Scientiae.

**VIÇOSA
MINAS GERAIS - BRASIL
2011**

VICTOR OLIVEIRA HERMSDORF

**MODELAGEM DA ATIVIDADE DE ELICITAÇÃO DO PROCESSO DE
ENGENHARIA DE REQUISITOS: UMA ABORDAGEM
UTILIZANDO DINÂMICA DE SISTEMAS**

Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do Programa de
Pós-Graduação em Ciência da
Computação, para obtenção do título
de *Magister Scientiae*.

APROVADA: 04 de fevereiro de 2011.

Prof. Alcione de Paiva Oliveira

Prof. Aziz Galvão da Silva Júnior

Prof. José Luis Braga
(Orientador)

*Dedico essa dissertação aos meus pais
Mozart e Eliene*

AGRADECIMENTOS

Agradeço primeiramente a Deus pelas oportunidades e pessoas especiais colocadas no meu caminho.

Agradeço aos meus pais, Mozart e Eliene, e às minhas irmãs Juliana e Marina por todo amor e incentivo.

Aos meus avós, pelo carinho.

Um agradecimento especial à minha namorada Akemí, que esteve ao meu lado durante toda a minha caminhada, apoiando, ouvindo, aconselhando, dando amor, carinho e atenção. Sem o seu apoio, essa conquista não seria possível.

Agradeço ao meu orientador, Professor José Luis Braga, por todo o conhecimento compartilhado. Com ele aprendi muito, ouvi conselhos que levarei por toda a vida. Mais que um orientador, o "Zé" foi sempre um amigo.

Agradeço ao meu co-orientador, Professor Moisés Resende Filho por dedicar o seu tempo para me auxiliar na pesquisa. Agradeço também os Professores Alcione de Paiva Oliveira e Aziz Galvão da Silva Júnior, por aceitarem o convite para contribuir com o meu trabalho através da participação da banca de dissertação.

A Universidade Federal de Viçosa, professores e funcionários do Departamento de Informática, por todo apoio necessário para que esse trabalho pudesse ser realizado. Um agradecimento especial ao secretário da pós-graduação Altino, que sempre me auxiliou com muito boa vontade.

A Fapemig, pelo apoio financeiro.

Agradeço aos meus amigos da república Cabrahmama, com os quais passei ótimos momentos ao longo dos últimos cinco anos.

Aos colegas do mestrado e da graduação, pelo apoio. Um agradecimento especial ao "Léo de Carvalho", amigo e companheiro de pesquisa, que muito contribuiu para que esse trabalho se realizasse.

Aos meus amigos e a todos que, de algum modo, contribuíram direta ou indiretamente para conclusão dessa etapa da minha vida.

BIOGRAFIA

VICTOR OLIVEIRA HERMSDORF, filho de Mozart Hermano Hermsdorf e Eliene Aparecida de Oliveira Hermsdorf, nascido em 13 de dezembro de 1985 na cidade de Ipatinga, Minas Gerais.

No ano de 2005, após concluir o curso técnico em Telecomunicações na cidade de Coronel Fabriciano, iniciou o curso de Ciência da Computação na Universidade Federal de Viçosa - UFV, concluído no ano de 2008.

Em 2009, ingressou no programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Viçosa, defendendo sua dissertação em fevereiro de 2011.

SUMÁRIO

SUMÁRIO	v
LISTA DE FIGURAS	vii
LISTA DE TABELAS	ix
RESUMO	x
ABSTRACT	xi
1 INTRODUÇÃO	1
1.1 O problema e sua importância	2
1.2 Objetivos	3
1.3 O fluxo de trabalho utilizado no desenvolvimento dessa pesquisa.....	4
1.4 Estrutura da dissertação	6
2 REFERENCIAL TEÓRICO.....	8
2.1 Dinâmica de sistemas.....	8
2.2 Engenharia de requisitos	12
2.3 Elicitação de requisitos	16
2.4 Trabalhos correlatos	19
3 MODELAGEM DA ATIVIDADE DE ELICITAÇÃO DE REQUISITOS 21	
3.1 Identificação e seleção de variáveis	22
3.2 Diagramas de influência	23
3.3 Diagrama estoque-fluxo.....	28
3.3.1 Taxa de Elicitação	28
3.3.2 Produtividade.....	33
3.3.3 Requisitos repetidos.....	35
3.3.4 Custo da elicitação.....	38
4 SIMULAÇÕES E ANÁLISES DOS RESULTADOS.....	40
4.1 Técnica de Entrevista.....	40
4.1.1 Cenário E1	40
4.1.2 Cenário E2	41
4.1.3 Cenário E3	42
4.1.4 Cenário E4.....	43
4.1.5 Cenário E5	45
4.1.6 Comparação dos cenários utilizando a técnica de entrevista.....	46
4.2 Comparação entre as técnicas de elicitação	50
4.2.1 Cenário C1	50
4.2.2 Cenário C2.....	57

5	PORTAL CORPORATIVO PARA APOIO A DECISÃO	65
6	CONCLUSÕES E TRABALHOS FUTUROS	71
6.1	Principais contribuições	72
6.2	Trabalhos futuros	72
	REFERÊNCIAS BIBLIOGRÁFICAS.....	74

LISTA DE FIGURAS

Figura 1.1. Custo da correção de defeitos de software	3
Figura 1.2. Fluxograma das etapas de desenvolvimento do trabalho	5
Figura 1.3. Mapa de conteúdos abordados no trabalho.....	7
Figura 2.1. Ilustração do exemplo de enchimento de uma banheira.....	10
Figura 2.2. Exemplo de diagrama de influência	11
Figura 2.3. Exemplo de diagrama de estoque-fluxo	11
Figura 2.4. Estado dos requisitos no trabalho de AMBRÓSIO (2008).....	20
Figura 3.1. Visão simplificada do modelo da fase de requisitos.....	21
Figura 3.2. Exemplo de GQM utilizado no trabalho.....	23
Figura 3.3. Relações dos elicitadores	24
Figura 3.4. Relações dos <i>stakeholders</i>	25
Figura 3.5. Relações da técnica de elicitação.....	25
Figura 3.6. Relações da produtividade.....	26
Figura 3.7. Relações de custos	27
Figura 3.8. Diagrama de influência da atividade de elicitação	27
Figura 3.9. Fluxo principal do modelo.....	28
Figura 3.10. Modelagem da produtividade	34
Figura 3.11. Relação entre número de clientes entrevistados e o percentual de necessidades identificadas.....	36
Figura 3.12. Comparação entre os gráficos obtidos utilizando a função beta-binomial	37
Figura 3.13. Modelagem dos requisitos repetidos	37
Figura 3.14. Modelagem dos custos.....	38
Figura 4.1. Cláusulas elicítadas no Cenário E1.....	41
Figura 4.2. Custo por cláusula correta no Cenário E1	41
Figura 4.3. Cláusulas elicítadas no Cenário E2.....	42
Figura 4.4. Custo por cláusula correta no Cenário E2	42
Figura 4.5. Cláusulas elicítadas no Cenário E3.....	43
Figura 4.6. Custo por cláusula correta no Cenário E3	43
Figura 4.7. Cláusulas elicítadas no Cenário E4.....	44
Figura 4.8. Custo por cláusula correta no Cenário E4	44
Figura 4.9. Cláusulas elicítadas no Cenário E5.....	45
Figura 4.10. Custo por cláusula correta no Cenário E5	45
Figura 4.11. Comparação de cláusulas elicítadas.....	46
Figura 4.12. Comparação de cláusulas elicítadas corretamente.....	47
Figura 4.13. Comparação de custo total.....	47
Figura 4.14. Comparação de custo final por cláusula correta	48
Figura 4.15. Cláusulas elicítadas em C1 utilizando a técnica de Entrevista	50
Figura 4.16. Custo por cláusula correta em C1 utilizando a técnica de Entrevista....	51
Figura 4.17. Cláusulas elicítadas em C1 utilizando a técnica de <i>Laddering</i>	51
Figura 4.18. Custo por cláusula correta em C1 utilizando a técnica de <i>Laddering</i> ...	52
Figura 4.19. Cláusulas elicítadas em C1 utilizando a técnica de <i>Sorting</i>	52

Figura 4.20. Custo por cláusula correta em C1 utilizando a técnica de <i>Sorting</i>	53
Figura 4.21. Cláusulas elicitadas em C1 utilizando a técnica de Análise de Protocolo	53
Figura 4.22. Custo por cláusula correta em C1 utilizando a técnica de Análise de Protocolo	54
Figura 4.23. Comparação de cláusulas elicitadas em C1	55
Figura 4.24. Comparação de cláusulas elicitadas corretamente em C1	55
Figura 4.25. Comparação de custo final por cláusula correta em C1	56
Figura 4.26. Cláusulas elicitadas em C2 utilizando a técnica de Entrevista	57
Figura 4.27. Custo por cláusula correta em C2 utilizando a técnica de Entrevista....	57
Figura 4.28. Cláusulas elicitadas em C2 utilizando a técnica de <i>Laddering</i>	58
Figura 4.29. Custo por cláusula correta em C2 utilizando a técnica de <i>Laddering</i> ...	58
Figura 4.30. Cláusulas elicitadas em C2 utilizando a técnica de <i>Sorting</i>	59
Figura 4.31. Custo por cláusula correta em C2 utilizando a técnica de <i>Sorting</i>	59
Figura 4.32. Cláusulas elicitadas em C2 utilizando a técnica de Análise de Protocolo	60
Figura 4.33. Custo por cláusula correta em C2 utilizando a técnica de Análise de Protocolo	60
Figura 4.34. Comparação de cláusulas elicitadas em C2.....	61
Figura 4.35. Comparação de cláusulas elicitadas corretamente em C2	62
Figura 4.36. Comparação de custo final por cláusula correta em C2	62
Figura 4.37. Comparação das porcentagens de cláusulas elicitadas em C2.....	64
Figura 5.1. Tela inicial do portal corporativo	65
Figura 5.2. Tela de cadastro de empresas	66
Figura 5.3. Tela de projetos.....	67
Figura 5.4. Tela de cadastro de projetos	67
Figura 5.5. Tela principal de um projeto.....	68
Figura 5.6. Tela do painel de controle	69
Figura 5.7. Tela de resultados de simulações.....	70

LISTA DE TABELAS

Tabela 2.1. Elementos do modelo estoque-fluxo da dinâmica de sistemas	9
Tabela 3.1. Comparação entre as técnicas de eliciação baseada em (DIESTE; JURISTO, 2010)	29
Tabela 3.2. Resumo dos resultados dos experimentos descritos em (CORBRIDGE et al., 1994; BURTON et al., 1990)	32
Tabela 3.3. Resultados do experimento realizado em (BURTON et al., 1990).....	33
Tabela 4.1. Resumo dos cenários de entrevista.....	46

RESUMO

HERMSDORF, Victor Oliveira, M.Sc., Universidade Federal de Viçosa, fevereiro de 2011. **Modelagem da atividade de elicitação do processo de engenharia de requisitos: uma abordagem utilizando dinâmica de sistemas.** Orientador: José Luis Braga. Co-Orientador: Moisés de Andrade Resende Filho.

O desenvolvimento de software é uma atividade que envolve riscos, parte deles relacionados a problemas de decisão, principalmente no nível gerencial. São muitas variáveis dinâmicas interconectadas para serem gerenciadas, sendo indispensável o uso de ferramentas que auxiliem os gerentes em suas tomadas de decisão, identificando os riscos antes de eles se materializarem. Na engenharia de requisitos, os impactos das decisões gerenciais são ainda maiores, visto que os requisitos irão direcionar todas as iterações posteriores no projeto. O objetivo desse trabalho é produzir um modelo de dinâmica de sistemas que aborde a atividade de elicitação de requisitos, permitindo que gerentes possam utilizá-lo como uma ferramenta de apoio para auxiliá-los na análise e tomadas de decisão relativas aos projetos de software.

ABSTRACT

HERMSDORF, Victor Oliveira, M.Sc., Universidade Federal de Viçosa, february, 2011. **Modeling the activity of elicitation of requirements engineering process: an approach using systems dynamics.** Adviser: José Luis Braga. Co-Adviser: Moisés de Andrade Resende Filho.

Software development is an activity which involves risks, some of them related to decision problems, especially at the management level. There are many interconnected dynamic variables to be managed, being necessary to use tools that assist managers in their decision making, identifying risks before they materialize. In requirements engineering the impacts of management decisions are even greater, since the requirements will direct all subsequent iterations in the project. The purpose of this work is to produce a system dynamics model that addresses the requirements elicitation activity, allowing managers to use it as a support tool to assist them in analysis and decision making on software projects.

1 INTRODUÇÃO

Atualmente, o *software* está presente, explícita ou implicitamente em todos os aspectos da vida (PFLEEGER, 2004). Por exemplo, softwares são utilizados desde o controle de aparelhos celulares até o controle de aeronaves e do tráfego aéreo. Diante disso, o desenvolvimento de softwares melhores, mais rápidos e mais baratos se torna condição essencial para o bem-estar da sociedade.

Apesar disso, não é incomum ocorrerem relatos de projetos de software mal sucedidos, seja por falhas na entrega dos softwares dentro do prazo e custo estabelecidos, ou mesmo pela má qualidade dos sistemas fornecidos pelas empresas de software (KELLNER *et al.*, 1999). Esses relatos têm um impacto direto na Engenharia de Software, que é a área responsável por estudar processos e métodos de desenvolvimento que possibilitem que os sistemas sejam desenvolvidos com qualidade e atendam as necessidades de seus usuários. Segundo SOMMERVILLE (2007), a Engenharia de Software é uma disciplina de engenharia que é focada em todos os aspectos da produção de *software*, desde os estágios iniciais da especificação do sistema até a sua manutenção, depois que ele entrou em operação.

Na Engenharia de Software, várias técnicas, métodos e processos foram propostos para aumentar a produtividade dos desenvolvedores e a qualidade dos sistemas desenvolvidos. Porém, se comparado às outras áreas do processo de desenvolvimento de software, pouca atenção foi dada aos aspectos gerenciais do processo (ABDEL-HAMID; MADNICK, 1991). As metodologias de gerenciamento não melhoraram com o passar dos anos tanto quanto as tecnologias de software (AURUM; WOHLIN, 2003). Isso é um fator alarmante, visto que problemas de gerenciamento estratégico e relacionados a fatores humanos já foram citados como uma das principais causas de falhas em projetos de software (KORSON; VAISHNAVI, 1992; RODRIGUES; WILLIAMS, 1997).

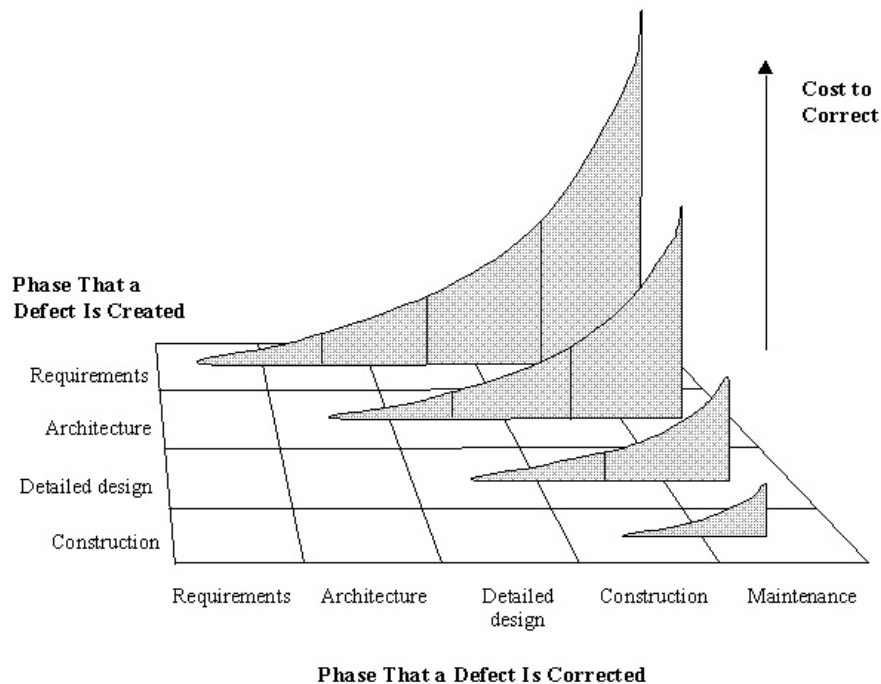
Dentro do contexto gerencial, problemas de decisão são muito comuns. Quantos profissionais alocar? Qual técnica utilizar? Qual o prazo necessário? Essas e muitas outras decisões são corriqueiras na vida dos gerentes de projetos.

As decisões gerenciais devem ser tomadas cuidadosamente, analisando-se todo o contexto e as conseqüências de cada possibilidade. Muitas vezes, porém, essas decisões se dão em cenários complexos e com muitas variáveis a serem analisadas, o que leva a tomadas de decisão reativas onde se considera apenas o problema presente sem relacioná-lo ao seu ambiente, suas variáveis e demais problemas correlacionados (AMBRÓSIO, 2008). Isso se torna um risco considerável no projeto, visto que decisões gerenciais tomadas de maneira equivocada podem comprometer todo um projeto levando a grandes prejuízos e, mais do que isso, podem levar à produção de softwares com falhas críticas, que podem significar grandes perdas no futuro, tanto financeiras como até mesmo de vidas humanas (PETERSON, 1996).

1.1 O problema e sua importância

Na extração de requisitos os impactos das decisões gerenciais são ainda maiores, visto que os requisitos formam uma base de trabalho que irá direcionar todas as iterações seguintes do desenvolvimento (AMBRÓSIO, 2008) e têm um grande impacto na qualidade final do software (DIESTE; LOPEZ; RAMOS, 2008). No processo de engenharia de requisitos é necessário desenvolver a compreensão dos requisitos o suficiente para permitir prosseguir com o projeto e construção em um nível de risco aceitável (WIEGERS, 2003). Um erro proveniente da especificação de requisitos detectado em uma fase avançada do desenvolvimento irá exigir retrabalho (SOMMERVILLE, 2007), com possibilidade de introdução de novos erros. Planos, modelos, esquemas, documentação deverão ser modificados para que a consistência e a rastreabilidade sejam mantidas (SAYÃO; LEITE, 2006). Se os requisitos não forem rastreáveis (RAMESH; JARKE, 2001) o problema é agravado, pois a dificuldade será maior para identificar todos os artefatos que precisam ser alterados (AMBRÓSIO, 2008).

Segundo (PRESSMAN, 2005), o custo de correção de erros na fase de projeto é cerca de três a seis vezes mais alto do que na fase de definição de requisitos. De acordo com (BOEHM; BASILI, 2001), é cem vezes mais caro reparar erros na produção do que durante as fases anteriores do desenvolvimento. MCCONNELL (1998) mostra que o custo para correção de um defeito criado na fase de requisitos cresce exponencialmente na medida em que o processo de desenvolvimento avança. Essa relação é mostrada através do gráfico na Figura 1.1.



Fonte: (MCCONNELL, 1998).

Figura 1.1. Custo da correção de defeitos de software

Dada a importância da engenharia de requisitos e a complexidade das relações entre as variáveis dinâmicas envolvidas nessa fase do processo de desenvolvimento de software, torna-se indispensável o uso de ferramentas que permitam enxergar o contexto como um todo (AMBRÓSIO, 2008). A dinâmica de sistemas (MADACHY, 2007; STERMAN, 2000) é uma dessas ferramentas, e será explorada neste trabalho para permitir melhor visualização do contexto e auxiliar na tomada de decisões sistêmicas.

1.2 Objetivos

O objetivo geral deste trabalho é produzir um modelo de dinâmica de sistemas que aborde a atividade de elicitação do processo de engenharia de requisitos de software, permitindo que gerentes possam utilizá-lo para auxiliá-los na análise e tomadas de decisão relativas aos projetos de software.

Os objetivos específicos relacionados são:

- Selecionar as principais variáveis dinâmicas envolvidas na atividade de elicitação;
- Elaborar diagramas de influência a partir da seleção de variáveis para explicitar como elas se relacionam;

- Mapear os diagramas de influência para um modelo estoque-fluxo da dinâmica de sistemas;
- Validar o modelo com dados da literatura que comprovem a sua consistência, e estabeleçam a viabilidade do seu uso no apoio à tomada de decisão;
- Construir uma ferramenta de apoio que permita o uso efetivo dos modelos de dinâmica de sistemas por empresas de software.

1.3 O fluxo de trabalho utilizado no desenvolvimento dessa pesquisa

Buscando alcançar os objetivos propostos neste trabalho, foram realizadas as atividades descritas a seguir. Um fluxograma das atividades é exibido na Figura 1.2.

1. Revisão bibliográfica: No início do trabalho, foi realizada uma extensa revisão bibliográfica para aprofundar os conhecimentos acerca do contexto em que o trabalho está inserido. Foram exploradas as áreas: dinâmica de sistemas, engenharia de requisitos e técnicas de elicitação de requisitos. Também foram investigados os trabalhos relacionados. O ponto de partida para o entendimento do contexto foi a leitura do livro “*Software Process Dynamics*” (MADACHY, 2007), da dissertação “*Modelagem da fase de requisitos em processos de desenvolvimento de software: uma abordagem utilizando dinâmica de sistemas*” (AMBRÓSIO, 2008) e de diversos artigos científicos que abordam assuntos relacionados ao trabalho desenvolvido. Esses trabalhos serão apresentados ao longo do texto.

2. Análise das variáveis dinâmicas da atividade de elicitação de requisitos: O segundo passo do trabalho foi realizar uma pesquisa aprofundada sobre engenharia de requisitos, com o objetivo de entender os aspectos dinâmicos envolvidos com a atividade de elicitação e com o processo de engenharia de requisitos de software. Nessa fase, as principais variáveis dinâmicas foram identificadas e selecionadas. Foram construídos diagramas de influência que ajudaram a entender o relacionamento entre as variáveis e a determinar os possíveis efeitos do tempo (dinamicidade) sobre essas relações.

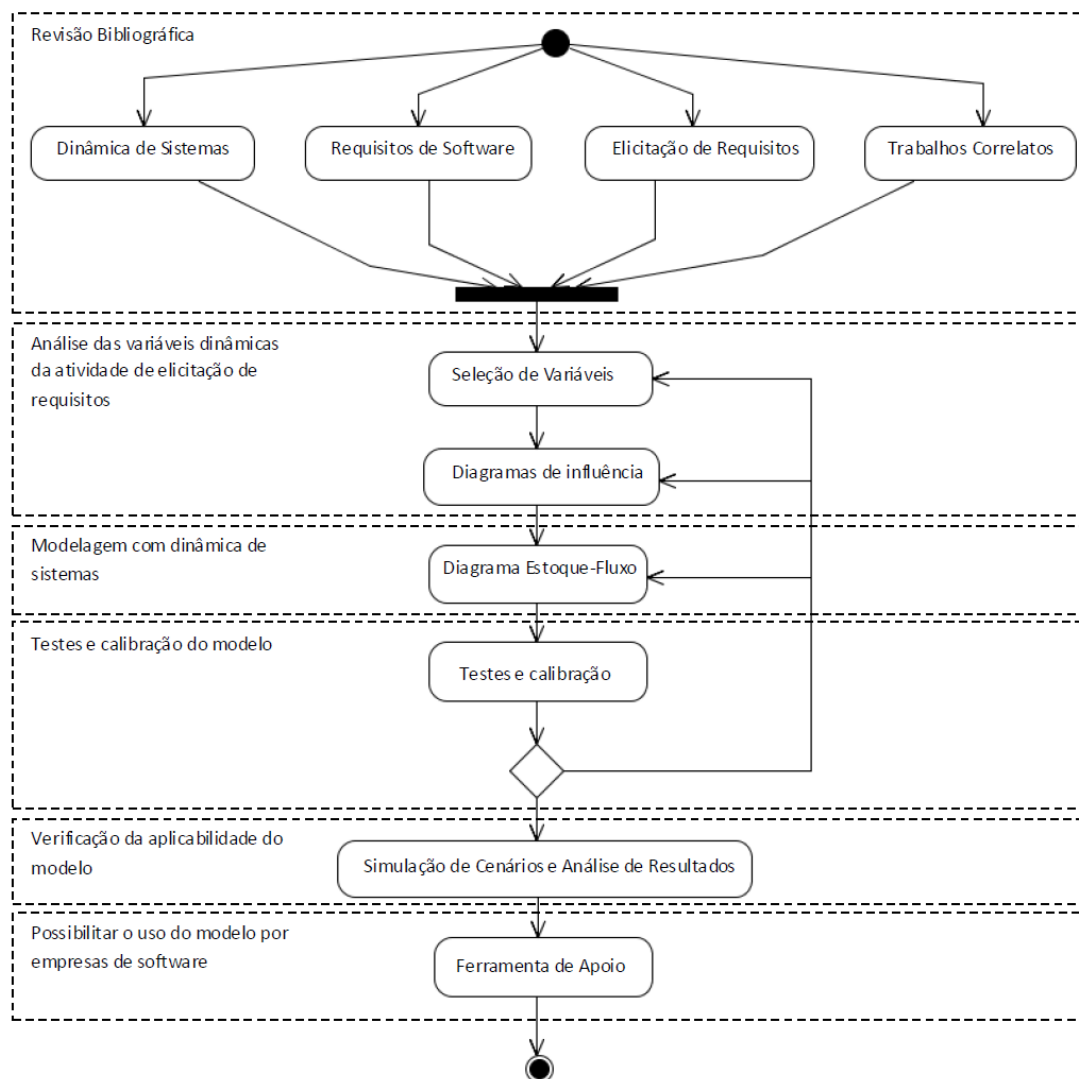
3. Modelagem com dinâmica de sistemas: Nesta fase foi realizado um mapeamento dos diagramas de influência obtidos no passo anterior em um diagrama de estoque-fluxo da dinâmica de sistemas.

4. Testes e calibração do modelo: Após a conclusão de cada versão do modelo, foram realizadas calibrações com dados disponíveis na literatura, sendo necessário

retornar aos passos anteriores para ajustar partes do modelo, corrigir ou ajustar relacionamentos e diagramas, incluir novas variáveis, etc. Esse fluxo foi repetido até que o modelo se tornou estável e condizente com a literatura.

5. Verificação da aplicabilidade do modelo: Foram elaborados cenários para verificar a aplicabilidade do modelo e a viabilidade do seu uso como uma ferramenta de apoio à tomada de decisão referente à atividade de elicitação de requisitos.

6. Possibilitar o uso do modelo por empresas de software: Para facilitar o acesso das empresas ao modelo, foi desenvolvida uma ferramenta de apoio, permitindo que simulações e análises sejam efetuadas remotamente, via Web.



Fonte: Elaborada pelo autor.

Figura 1.2. Fluxograma das etapas de desenvolvimento do trabalho

1.4 Estrutura da dissertação

No Capítulo 2 é apresentado o referencial teórico, resultado da revisão bibliográfica realizada. Nesse capítulo são discutidos os principais conceitos relacionados ao trabalho. O primeiro tema a ser abordado é a dinâmica de sistemas (Seção 2.1), ferramenta utilizada para modelar e realizar simulações. O contexto da modelagem é o processo de engenharia de requisitos (Seção 2.2), e mais especificamente a atividade de elicitação de requisitos (Seção 2.3), que é uma das atividades realizadas na engenharia de requisitos. Nesse capítulo são apresentados ainda os trabalhos correlatos (Seção 2.4), que foram tomados como base para o desenvolvimento deste trabalho.

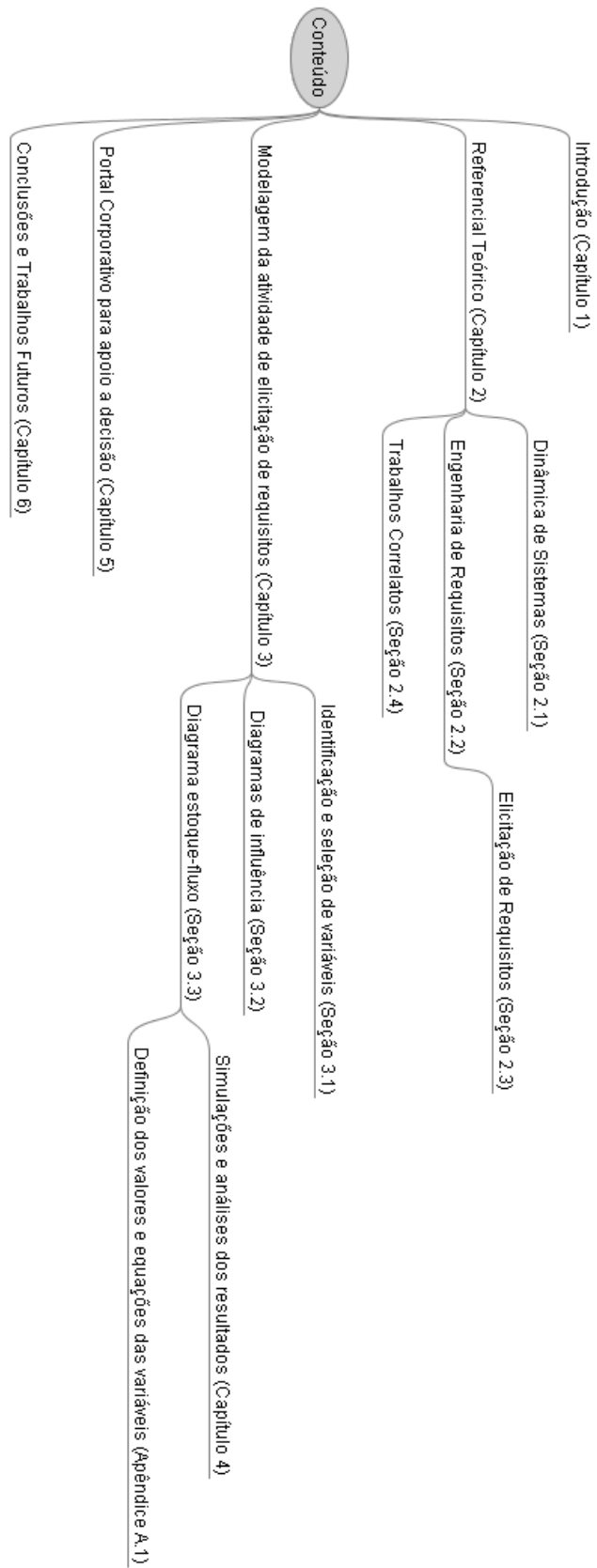
O Capítulo 3 descreve os modelos desenvolvidos. São descritos a seleção de variáveis (Seção 3.1), os diagramas de influência (Seção 3.2) e o diagrama estoque-fluxo (Seção 3.3). Além disso, são apresentados os trabalhos da literatura nos quais as relações entre as variáveis foram baseadas.

No Capítulo 4 são descritas algumas simulações realizadas e suas análises. Foram criados vários cenários que representam diferentes fases iniciais de projetos de software. Esses cenários facilitam as comparações das simulações e mostram como é possível obter informações relevantes a partir de simulações do modelo proposto.

O Capítulo 5 apresenta o Portal Corporativo do LES/UFV (Laboratório de Engenharia de Software da Universidade Federal de Viçosa), uma ferramenta de apoio disponibilizada para facilitar o uso efetivo dos modelos de simulação por gerentes de projetos de empresas de software.

Por fim, no Capítulo 6 são apresentadas as conclusões, principais contribuições deste trabalho e as sugestões de trabalhos futuros.

Um mapa da relação dos conteúdos abordados neste trabalho é apresentado na Figura 1.3.



Fonte: Elaborada pelo autor.

Figura 1.3. Mapa de conteúdos abordados no trabalho

2 REFERENCIAL TEÓRICO

2.1 Dinâmica de sistemas

Criada por Jay W. Forrester no MIT (*Massachusetts Institute of Technology*) na década de 1950 (LANE, 2006), a dinâmica de sistemas é uma forma de modelagem para simulação computacional que utiliza os conceitos de realimentação de informação e variáveis de estado para modelar sistemas e explorar a ligação entre a estrutura do sistema e o comportamento evolutivo no tempo (FORRESTER, 1961, 1968).

A dinâmica de sistemas é uma técnica descritiva (BRAGA *et al.*, 2004), baseada no pensamento e análise sistêmicos e na teoria matemática dos Sistemas Dinâmicos (STERMAN, 2000). Sua grande utilidade é permitir a compreensão de como as políticas adotadas, ou a própria estrutura do sistema, afetam ou determinam o seu comportamento dinâmico (STERMAN, 2000), antecipando colapsos (AMBRÓSIO, 2008).

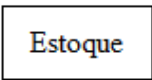

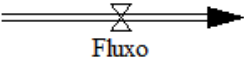
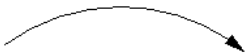
Uma fase inicial da modelagem normalmente é desenvolvida utilizando-se os diagramas de influência (SENGE, 1990). Diagramas de influência (*causal loop diagrams*) são diagramas simples com palavras e setas que ajudam a descrever relações de causa e efeito e realimentação de informações em um sistema (MADACHY, 2007). Nesses diagramas, as setas possuem sinais positivo (+) e negativo (-) para representar o relacionamento entre as variáveis ligadas por elas. As setas positivas (+) indicam que as variáveis de causa e efeito (início e ponta da seta, respectivamente) variam na mesma direção, ou seja, com tudo mais permanecendo constante, se a causa aumenta (diminui), o efeito aumenta (diminui). A seta negativa (-) indica que causa e efeito variam em direções opostas, se a causa aumenta (diminui), o efeito diminui (aumenta) (STERMAN, 2000).

Os diagramas de influência permitem visualizar graficamente as relações entre as variáveis chaves do problema, permitindo localizar possíveis laços de realimentação e atrasos nos efeitos resultantes das interações entre essas variáveis

(BRAGA *et al.*, 2004). Apesar disso, os diagramas de influência muitas vezes se mostram muito simplistas ou incompletos para explicar um sistema e seu comportamento (MADACHY, 2007; RICHARDSON, 1986). É necessário então fazer uso de diagramas mais poderosos, que possuem mais recursos para melhor representar sistemas e comportamentos. São utilizados então os diagramas estoque-fluxo.

Os elementos presentes nos diagramas estoque-fluxo são descritos na Tabela 2.1. A notação utilizada na tabela é referente à ferramenta Vensim (VENSIM, 2010).

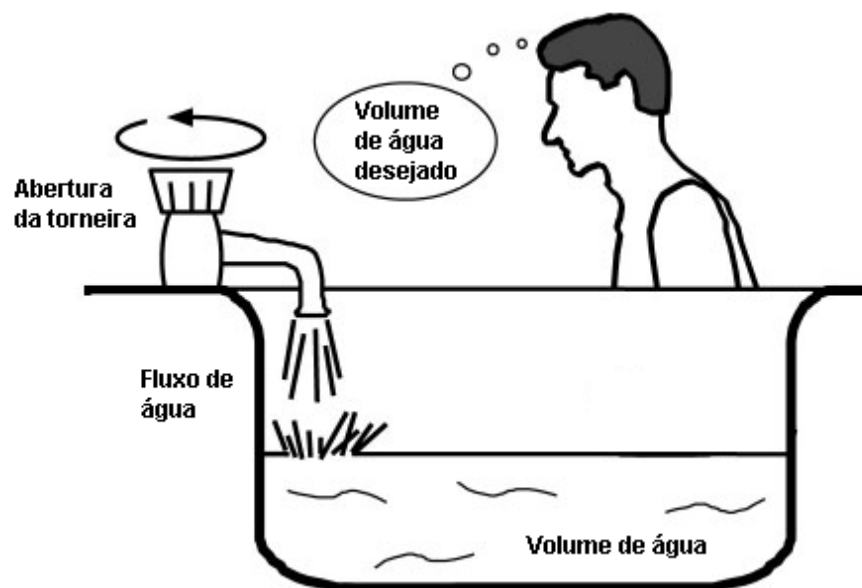
Tabela 2.1. Elementos do modelo estoque-fluxo da dinâmica de sistemas

Elemento	Notação	Descrição
Estoque (<i>Level</i>)		<i>Estoques</i> são acúmulos ao longo do tempo. São funções de acumulações passadas vindas de <i>fluxos</i> .
Fonte/Sumidouro (<i>Source/Sink</i>)		<i>Fontes</i> ou <i>sumidouros</i> indicam que os fluxos vêm ou vão para algum lugar externo ao processo. Eles representam fontes infinitas ou repositórios que não são especificados no modelo.
Fluxo (<i>Rate</i>)		<i>Fluxos</i> são as "ações" no sistema. Eles são responsáveis pelas mudanças nos <i>estoques</i> (acúmulo ou uso de recursos) e podem representar as decisões e declarações políticas.
Variável Auxiliar (<i>Auxiliary</i>)	<i>variável auxiliar</i>	<i>Auxiliares</i> são os conversores de entrada para saída e ajudam a elaborar detalhes da estrutura de estoque e fluxo. Eles exercem influência sobre os valores dos <i>fluxos</i> através dos <i>elos de informação</i> .
Elo de informação (<i>Information Link</i>)		<i>Elos de informação</i> são usados para representar fluxos de informação (em vez de fluxo de material, como no caso dos <i>fluxos</i>). Eles podem representar loops de realimentação entre os elementos. Eles ligam parâmetros do processo aos <i>fluxos</i> e <i>variáveis auxiliares</i> .

Fonte: Adaptada de (MADACHY, 2007)

Segundo (LANE, 2008), as principais vantagens da utilização de diagramas de estoque-fluxo são: foco em entidades, recursos e atributos mensuráveis; distinção entre estoque e fluxo e entre fluxos e elos de informação; são melhores para mostrar os processos causais reais sendo modelados, e como operam os diferentes loops no processo; forçam uma especificação mais detalhada dos mecanismos causais; fornecem uma base melhor para inferir comportamentos; e são diagramas prontos para simulação.

Para ilustrar o uso de diagramas de influência e diagramas estoque-fluxo é apresentado na Figura 2.1 um exemplo clássico na literatura: o enchimento de uma banheira. Uma pessoa enche uma banheira através da abertura de uma torneira. No início, ela abre totalmente a torneira para que o fluxo de água seja maior e aumente mais rápido o volume de água na banheira. Na medida em que o volume de água na banheira aumenta, a pessoa vai fechando a torneira para evitar que a água transborde, fazendo com que a água chegue somente até o volume desejado.

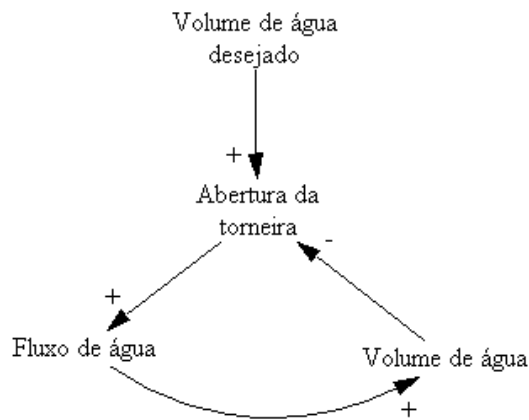


Fonte: Adaptada de (LANE, 2008)

Figura 2.1. Ilustração do exemplo de enchimento de uma banheira

A situação ilustrada na Figura 2.1 pode ser descrita através de um diagrama de influência, conforme apresentado na Figura 2.2. Quanto maior é o volume de água desejado, maior é a abertura da torneira (sinal +). Quanto mais a torneira é aberta, mais aumenta o fluxo de água que sai da torneira (sinal +), que, por sua vez aumenta o volume de água na banheira (sinal +). Quanto maior o volume de água na banheira, mais esse volume se aproxima do volume desejado. Dessa forma, quanto maior o

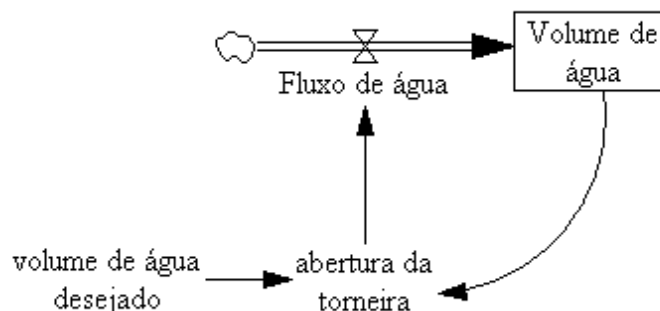
volume de água, menor deve ser a abertura da torneira (sinal -), para evitar que o volume de água ultrapasse o volume desejado e transborde.



Fonte: Adaptada de (LANE, 2008)

Figura 2.2. Exemplo de diagrama de influência

A situação ilustrada na Figura 2.1 pode também ser representada por meio de um diagrama de estoque-fluxo, conforme apresentado na Figura 2.3. Estão representados o estoque “Volume de água”, o fluxo “Fluxo de água” e as variáveis auxiliares “volume de água desejado” e “abertura da torneira”. O valor do estoque “Volume de água” é modificado somente pelo fluxo “Fluxo de água”, que é influenciado pela variável auxiliar “abertura da torneira”. A variável “abertura da torneira” por sua vez, é influenciada pela variável “volume de água desejado” e pelo valor do estoque “Volume de água”.



Fonte: Adaptada de (LANE, 2008)

Figura 2.3. Exemplo de diagrama de estoque-fluxo

Uma das desvantagens de se utilizar somente o diagrama de influência é percebida no exemplo apresentado. A relação positiva entre “fluxo de água” e “volume de água” indica que as duas variam “na mesma direção”. Porém, essa

relação pode levar a um entendimento errôneo do sistema. Pode parecer que se o fluxo de água diminuir, o volume de água na banheira também irá diminuir, o que não condiz com a realidade. Nesse caso, utilizando o diagrama de estoque-fluxo é possível verificar que o “Fluxo de água” é somente um fluxo de entrada para o estoque “Volume de água”. Dessa forma somente será possível aumentar o valor do estoque através desse fluxo, ou seja, nunca será possível reduzi-lo.

A dinâmica de sistemas é uma técnica de modelagem que não está diretamente associada a nenhuma disciplina ou área de conhecimento específica (RICHMOND, 2001), podendo ser utilizada para modelar sistemas com comportamento dinâmico de qualquer domínio de conhecimento (AMBRÓSIO, 2008).

No caso da modelagem de processos de desenvolvimento de software, a dinâmica de sistemas provê apoio através da investigação dos aspectos em níveis micro e macro. No nível micro, ela pode ser usada para comparar o ciclo de vida de diferentes processos, técnicas de detecção de erro, casos de negócio, decidir o “quanto é o bastante” em termos de testes ou outros problemas. Já no nível macro, organizações podem focar em aspectos específicos do custo de desenvolvimento, cronograma, qualidade do produto, entre outras questões (MADACHY, 2007).

2.2 Engenharia de requisitos

O sucesso de um software depende do quanto ele se adéqua às necessidades dos usuários e ao seu ambiente (NUSEIBEH; EASTERBROOK, 2000; PARNAS, 1999). Requisitos de software expressam essas necessidades (CHENG; ATLEE, 2007), assim como as limitações dos softwares, que têm como objetivo contribuir para a solução de um problema do mundo real (KOTONYA; SOMMERVILLE, 2000).

Os requisitos de um sistema são as descrições dos serviços prestados pelo sistema e suas restrições operacionais. O processo de descobrir, analisar, documentar e validar esses serviços e restrições é chamado engenharia de requisitos (SOMMERVILLE, 2007).

Um dos principais objetivos da engenharia de requisitos é melhorar a modelagem de sistemas e a capacidade de analisá-los, possibilitando maior entendimento de suas características antes da implementação (ROBINSON *et al.*,

2003). Segundo SOMMERVILLE (2007), o processo global de engenharia de requisitos inclui quatro sub-processos: avaliar se o sistema é realmente útil para o negócio (estudo de viabilidade); descobrir requisitos (elicitação e análise); escrever os requisitos em algum formulário padrão (especificação); e verificar se os requisitos realmente definem o sistema que o usuário necessita (validação).

Para iniciar qualquer sistema, o processo da engenharia de requisitos deve começar pelo estudo de viabilidade. A base para esse estudo é um conjunto de requisitos do negócio preliminar e um rascunho da descrição do sistema e como o sistema destina-se a dar suporte ao processo do negócio. O resultado desse estudo deve ser um relatório recomendando se é válido ou não continuar com a engenharia de requisitos e com o processo de desenvolvimento do sistema (SOMMERVILLE, 2007).

A elicitação de requisitos é a atividade responsável por descobrir a origem dos requisitos e como os elicitadores¹ podem coletá-los (SWEBOK, 2004). Nessa etapa, os elicitadores coletam as informações sobre o domínio do problema (DIESTE; JURISTO, 2010). Na próxima seção serão fornecidos maiores detalhes sobre essa atividade.

Na atividade de análise, os requisitos são organizados, classificados e priorizados, agrupando requisitos relacionados e organizando-os em conjuntos coerentes (SOMMERVILLE, 2007). É realizada também a modelagem conceitual, onde vários tipos de modelos são desenvolvidos, como fluxo de controle, modelos de estado, interações do usuário, modelos de objeto, entre outros; e o mapeamento dos requisitos para cada componente do sistema. Dessa forma, são descobertos os limites do software e como ele deve interagir com seu ambiente (SWEBOK, 2004). Além disso, quando ocorrem conflitos entre requisitos mutuamente incompatíveis, é função da atividade de análise resolver esses conflitos através da negociação (SOMMERVILLE, 2007; SWEBOK, 2004).

A especificação de requisitos se refere à produção de um ou mais documentos de requisitos que podem ser sistematicamente revisados, avaliados e aprovados (SWEBOK, 2004). Os requisitos devem ser descritos com precisão suficiente para

¹ Elicitadores são os profissionais que efetivamente realizam a elicitação. Na literatura esses profissionais recebem diversas denominações, como engenheiros de requisitos, especialistas de requisitos e engenheiros de software.

que seja possível validá-los, verificar a sua implementação e estimar o seu custo (SWEBOK, 2004). Em sistemas maiores e mais complexos, particularmente envolvendo outros componentes que não são softwares, são utilizados até três documentos na atividade de especificação: documento de definição do sistema, especificação de requisitos do sistema e especificação de requisitos do software (SWEBOK, 2004). O documento de definição do sistema, para o qual o IEEE (*Institute of Electrical and Electronics Engineers*) estabelece padrões para sua formulação através da norma IEEE 1362 (IEEE1362, 1998), define os requisitos em alto nível, da perspectiva do domínio. A especificação de requisitos do sistema, cujo guia para sua formulação é o IEEE 1233 (IEEE1233, 1998), especifica o sistema como um todo, além do escopo do software, como por exemplo, um sistema para aviões modernos, onde existem diferentes hardwares envolvidos. A especificação de requisitos do software, cujo padrão segue o IEEE 830 (IEEE830, 1998), estabelece a base de acordo entre clientes e prestadores de serviço do que o produto de software deve ou não fazer. Em sistemas de software comuns, apenas o documento de especificação de requisitos do software é necessário (SWEBOK, 2004).

Por fim, os documentos produzidos na atividade de especificação são submetidos à validação e verificação, que podem ser feitas através da revisão dos documentos de requisitos, onde os requisitos são analisados sistematicamente pela equipe de revisores; prototipação, em que um modelo executável do sistema é apresentado para os usuários finais e clientes; validação dos modelos, onde é verificada a qualidade dos modelos desenvolvidos na atividade de análise; geração de casos de testes, em que são planejados testes para cada um dos requisitos; e testes de aceitação, que verifica se o produto acabado satisfaz cada um dos requisitos (SOMMERVILLE, 2007; SWEBOK, 2004). A validação de requisitos é extremamente importante para evitar que erros provenientes da especificação sejam propagados para as fases posteriores do desenvolvimento, aumentando consideravelmente o retrabalho e, conseqüentemente, o custo do projeto (SOMMERVILLE, 2007; PRESSMAN, 2005; BOEHM; BASILI, 2001; MCCONNELL, 1998).

A engenharia de requisitos é focada na definição precisa do problema que o software deve resolver (ou seja, definir o que o software deve fazer), enquanto outras áreas da engenharia de software são focadas na definição e refinamento da solução

de software proposta. Segundo CHENG; ATLEE (2007), essa diferença torna a engenharia de requisitos inerentemente difícil por uma séria de fatores:

- Elicitadores começam com idéias mal definidas e muitas vezes conflitantes sobre o que o sistema proposto deve fazer, e devem progredir para um documento técnico único e detalhado da especificação do sistema.
- O domínio do problema de requisitos é menos restrito do que o domínio da solução do software – na verdade, é a definição dos requisitos que auxilia a delimitar o domínio da solução. Como tal, há muito mais opções para se considerar e decisões para se tomar a respeito dos requisitos, como selecionar e priorizar os requisitos, decidir sobre os limites do sistema, negociar a resolução de conflitos de requisitos, definir os critérios de aceitação, e assim por diante (LAMSWEERDE, 2000).
- Uma forma de simplificar o domínio do problema é restringir as condições do ambiente sob as quais o sistema deverá operar. Em tais casos, raciocinar sobre os requisitos envolve o raciocínio sobre a combinação do comportamento do sistema proposto e das suposições feitas sobre o ambiente do sistema. Levar em consideração as condições do ambiente aumenta significativamente a complexidade do problema em questão, uma vez que o ambiente de um sistema pode ser uma combinação de dispositivos de hardware, fenômenos do mundo físico, o comportamento humano (operador ou usuário), e outros componentes de software.
- O raciocínio sobre o ambiente do sistema inclui a identificação não apenas das suposições sobre o comportamento normal do ambiente, mas também sobre as possíveis ameaças ou riscos que o ambiente pode representar para o sistema.
- Os artefatos de requisitos resultantes têm de ser entendidos por especialistas e todos os interessados no sistema, que podem não ter conhecimento sobre computação. Assim, as notações de requisitos e processos devem manter um delicado equilíbrio entre a produção de descrições para pessoas que não são da área de computação e a produção de documentos técnicos que são suficientemente precisos para a equipe de desenvolvedores.

Um processo de engenharia de requisitos bem executado é essencial para o sucesso de projetos de software, pois garante que o software a ser desenvolvido vai de encontro às necessidades dos clientes. Quando o software é desenvolvido para um

mercado geral, em vez de ser para um cliente específico, existe uma pressão maior, dada a necessidade de lançar o produto o mais rápido possível. Uma engenharia de requisitos efetiva é um importante fator de sucesso para conseguir alcançar a demanda do mercado (HÖST *et al.*, 2001).

Os aspectos gerenciais do processo de engenharia de requisitos são cruciais para o êxito do desenvolvimento de produtos de software (AURUM; WOHLIN, 2003). O gerenciamento é um processo no qual os gerentes asseguram que os recursos são obtidos e utilizados de maneira eficaz e eficiente na realização dos objetivos da organização (ANTHONY, 1965). Apesar de crucial, o gerenciamento não é uma tarefa trivial na engenharia de requisitos, visto que é um processo complexo de comunicação e negociação envolvendo muitos profissionais (clientes, arquitetos de software, gerentes de projetos e mantenedores do sistema), que devem decidir o que fazer, quando fazer, quais informações são necessárias e quais ferramentas utilizar (DAVIS, 1990; SOMMERVILLE; SAWYER, 1997).

Embora a maioria dos modelos de processos de engenharia de requisitos e métodos de especificação tenha sido criada por pessoas que gostam de precisão e diagnóstico, na realidade o gerenciamento de requisitos ainda é um processo político. A maioria dos projetos de software vem das preferências dos gestores, das suas intuições, ou de seus projetos passados (ANDRIOLE, 1998). STRIGINI (1996) enfatiza que na indústria de software, em muitos casos, decisões importantes são subjetivas. Em alguns casos, a subjetividade é inevitável, mas é importante que os desenvolvedores reconheçam os perigos e tomem precauções quanto às conseqüências.

2.3 Elicitação de requisitos

A elicitação de requisitos é o primeiro estágio no entendimento do problema que o software deve resolver, sendo uma atividade fundamentalmente humana (SWEBOK, 2004) e podendo envolver várias pessoas na organização. O termo *stakeholder* é utilizado para referenciar todas as pessoas afetadas direta ou indiretamente pelo sistema, podendo incluir os usuários finais que irão interagir com o sistema, gerentes do negócio, especialistas no domínio, etc. (SOMMERVILLE, 2007).

Na elicitação, os *stakeholders* são identificados e são estabelecidos os relacionamentos iniciais entre clientes e equipe de desenvolvimento (SWEBOK, 2004). Um dos fatores-chave para o sucesso dessa atividade é a boa comunicação entre os *stakeholders* e os elicitadores.

Identificar os requisitos do sistema é um dos maiores desafios encontrados na produção de software. A elicitação de requisitos é talvez a tarefa de desenvolvimento de software mais difícil, propensa a erros e mais crítica (WIEGERS, 2003), uma vez que a elicitação mal feita irá quase garantir um fracasso completo do projeto (HICKEY; DAVIS, 2003). De acordo com BROOKS (1995), é impossível para os *stakeholders* especificar completamente, de forma precisa e corretamente os requisitos de um produto de software antes de ter uma primeira versão do produto construída. SOMMERVILLE (2007) aponta várias razões pelas quais é difícil elicitar requisitos dos *stakeholders*:

- *Stakeholders* geralmente não sabem o que eles querem do sistema, exceto pelos termos mais gerais.
- *Stakeholders* naturalmente expressam os requisitos utilizando seus próprios termos e com conhecimento implícito dos seus próprios trabalhos.
- Diferentes *stakeholders* possuem requisitos diferentes, que podem expressar de diferentes maneiras.
- Fatores políticos podem influenciar os requisitos de um sistema. Por exemplo, gerentes podem demandar requisitos que irão aumentar a sua influência na organização.
- O ambiente econômico e empresarial é dinâmico. Novos requisitos podem surgir a partir de novos *stakeholders* que nem foram consultados a princípio.

Para contornar as dificuldades da elicitação de requisitos podem ser utilizadas diversas técnicas de elicitação. Essas técnicas são mecanismos de interações entre elicitadores e *stakeholders* (DIESTE; JURISTO, 2010), onde o objetivo é extrair dos *stakeholders* o máximo de informações possível sobre o domínio do problema que o software deve resolver. Neste trabalho, serão consideradas as técnicas de entrevista (JOHNSON; JOHNSON, 1987), análise de protocolo (ERICSON; SIMON, 1984), *sorting* (GAMMACK, 1987) e *laddering* (LAFRANCE, 1986; CORBRIDGE *et al.*, 1994).

A entrevista é a técnica mais comumente utilizada na prática (DAVIS *et al.*, 2006). Ela consiste em diálogos entre elicitadores e *stakeholders* sobre o domínio do problema no qual o software deve atuar. Existem diversos tipos de entrevistas, como entrevistas estruturadas (mais formais, sendo preparada com antecedência, onde o entrevistador faz as mesmas questões e na mesma ordem para cada um dos entrevistados), não-estruturadas (diálogo menos formal, onde não há um roteiro rígido a ser seguido, dando liberdade ao entrevistado de abordar os assuntos a seu próprio modo) e grupo foco (onde mais pessoas participam da entrevista formando um grupo de discussão). Neste trabalho será abordada a técnica de entrevista não-estruturada.

Na técnica de análise de protocolo, o *stakeholder* "pensa alto" enquanto resolve o problema que o software deve dar suporte (DIESTE; JURISTO; SHULL, 2008). Enquanto isso o elicitador acompanha o raciocínio do *stakeholder* e faz suas anotações.

Na técnica de *sorting*, o *stakeholder* agrupa conceitos similares e entidades do domínio do problema. Cada conceito ou entidade é impresso em um cartão para que o *stakeholder* possa fisicamente organizar e reorganizar para ajudar a categorizar o domínio do problema (DIESTE; JURISTO; SHULL, 2008).

No *laddering*, as informações são organizadas em uma hierarquia dos elementos do domínio do problema. O elicitador pede que o *stakeholder* comece falando sobre um conceito "raiz" do problema. Depois são feitas perguntas curtas e diretas que ajudam a explicar a importância do conceito no domínio (DIESTE; JURISTO; SHULL, 2008) e que permitem "caminhar" na estrutura hierárquica do problema. Por exemplo, dada uma classe de problemas, para "descer" na estrutura pode ser perguntado "É possível fornecer exemplos dessa classe?". Ao contrário, para "subir" na hierarquia pode ser perguntado "Os itens A e B são exemplos de qual classe de problema?". Existe ainda a possibilidade de "caminhar lateralmente" na hierarquia, perguntado "Quais outros exemplos dessa classe podem ser fornecidos além do item A?". Perguntas adicionais como "Qual a principal diferença entre os itens A e B?" são usadas para elicitare atributos dos conceitos na hierarquia (CORBRIDGE *et al.*, 1994).

A escolha da técnica de elicitação de requisitos depende do tempo e recursos disponíveis para o projeto, e também do tipo de informação que necessita ser extraída (NUSEIBEH; EASTERBROOK, 2000). No escopo deste trabalho, mais importante do que estudar cada técnica específica, é a comparação entre as técnicas no que diz respeito à efetividade, qualidade e tempo gasto na elicitação.

2.4 Trabalhos correlatos

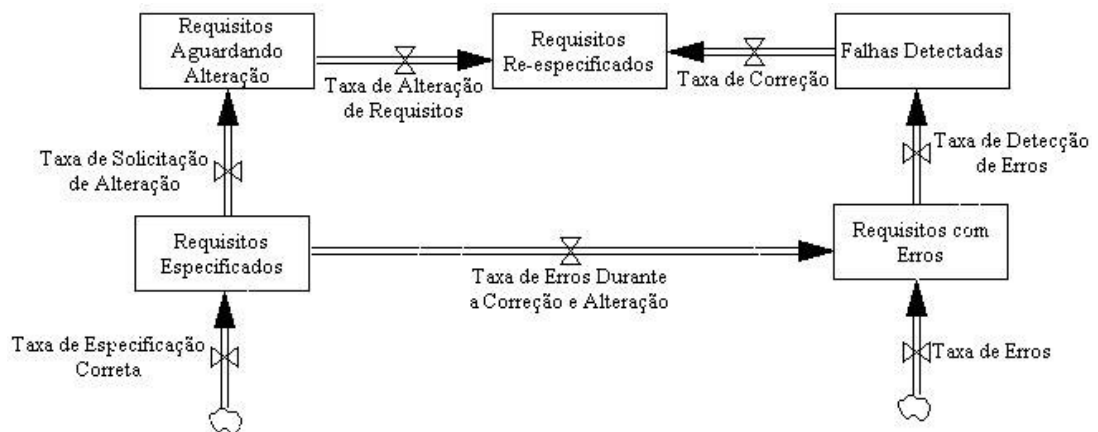
Entre os trabalhos que comparam as técnicas de elicitação e as principais variáveis que influenciam durante a atividade de elicitação de requisitos, destacam-se os trabalhos (DIESTE; JURISTO, 2010), (DIESTE; LOPEZ; RAMOS, 2008) e (DAVIS *et al.*, 2006). Esses trabalhos realizaram uma agregação de vários estudos sobre as técnicas de elicitação, onde diversas técnicas foram comparadas e os resultados dos estudos foram fornecidos de maneira resumida através de tabelas comparativas. As comparações descritas nesses trabalhos foram todas qualitativas, ou seja, mostraram quais técnicas eram superiores às outras em vários quesitos, porém não informaram “o quanto” eram superiores. Dessa forma, esses trabalhos foram tomados como ponto de partida para a comparação entre as técnicas de elicitação e também serviram como validação para os resultados que foram obtidos nessa pesquisa.

Entre os diversos trabalhos relacionados à aplicação de dinâmica de sistemas na modelagem de processos de desenvolvimento de software, destacam-se os trabalhos (ABDEL-HAMID; MADNICK, 1991) e (MADACHY, 2007). Em relação à modelagem focada em requisitos de software, destacam-se os trabalhos (AMBRÓSIO, 2008) e (AMBRÓSIO *et al.*, 2008).

AMBRÓSIO (2008) propôs um modelo de dinâmica de sistemas para analisar efeitos mútuos de variáveis envolvidas na fase de requisitos do processo de desenvolvimento de software. Através desse modelo, gerentes de projeto podem buscar alternativas para avaliar o risco de suas decisões, fornecendo parâmetros que caracterizam sua equipe e executando simulações, com o intuito de verificar as possíveis conseqüências de suas decisões. São abordados no modelo temas como o *turnover* de pessoal, retrabalho devido a erros, falhas em planejamento, entre outros.

Considerando-se como atividades da fase de requisitos a elicitação, análise, especificação e validação (SWEBOK, 2004; SOMMERVILLE, 2007), o modelo de

AMBRÓSIO (2008) é focado nas atividades de especificação e validação de requisitos. Não são tratadas no modelo as atividades anteriores à especificação. Considera-se que os requisitos do software vêm de um escopo externo. A Figura 2.4 mostra os estoques referentes aos estados dos requisitos durante a simulação no trabalho de AMBRÓSIO (2008). As atividades realizadas no decorrer da especificação dos requisitos são responsáveis por acrescentar requisitos nesses estoques e transferir requisitos de um estoque para o outro, alterando o estado dos requisitos (AMBRÓSIO, 2008).



Fonte: (AMBRÓSIO, 2008)

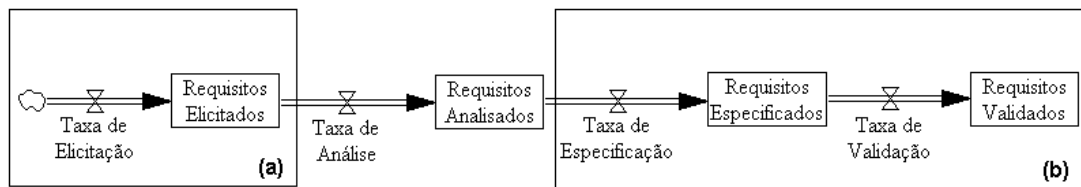
Figura 2.4. Estado dos requisitos no trabalho de AMBRÓSIO (2008)

Um complemento necessário ao modelo proposto por AMBRÓSIO (2008) é a contextualização dos requisitos que alimentam o modelo nas atividades de elicitação e análise. No presente trabalho, será abordada a contextualização dos requisitos na atividade de elicitação, onde serão analisadas as variáveis dinâmicas que impactam nessa atividade e que devem ser levadas em consideração para as tomadas de decisão nessa etapa do processo da engenharia de requisitos. Apesar de ser um complemento para o modelo de Ambrósio (2008), o modelo produzido neste trabalho é totalmente independente.

3 MODELAGEM DA ATIVIDADE DE ELICITAÇÃO DE REQUISITOS

Para contextualizar o presente trabalho, é importante apresentar uma visão geral do modelo que será produzido com a junção dos trabalhos na área. A Figura 3.1 mostra um modelo simplificado dessa visão.

Para construir os modelos apresentados nesse trabalho foi utilizada a ferramenta Vensim (VENSIM, 2010), em sua versão gratuita para uso acadêmico.



Fonte: Elaborada pelo autor.

Figura 3.1. Visão simplificada do modelo da fase de requisitos

Partindo do modelo apresentado na Figura 3.1, o objetivo geral dessa linha de pesquisa é estudar os fatores que influenciam cada “taxa” apresentada no modelo. É importante ressaltar que esse modelo é uma versão simplificada, portanto não são exibidas, por exemplo, as taxas de erros e retrabalhos, apesar de essas relações serem de extrema importância no estudo da área.

O trabalho de AMBRÓSIO (2008) estuda o contexto destacado pelo retângulo (b) na Figura 3.1, como já foi discutido na seção de trabalhos correlatos.

O modelo de dinâmica de sistemas apresentado neste trabalho abrange as principais variáveis envolvidas na atividade de elicitação e alguns dos relacionamentos existentes entre elas. As variáveis foram definidas e quantificadas de acordo com dados disponíveis na literatura. Na Figura 3.1, o contexto deste trabalho é destacado pelo retângulo (a).

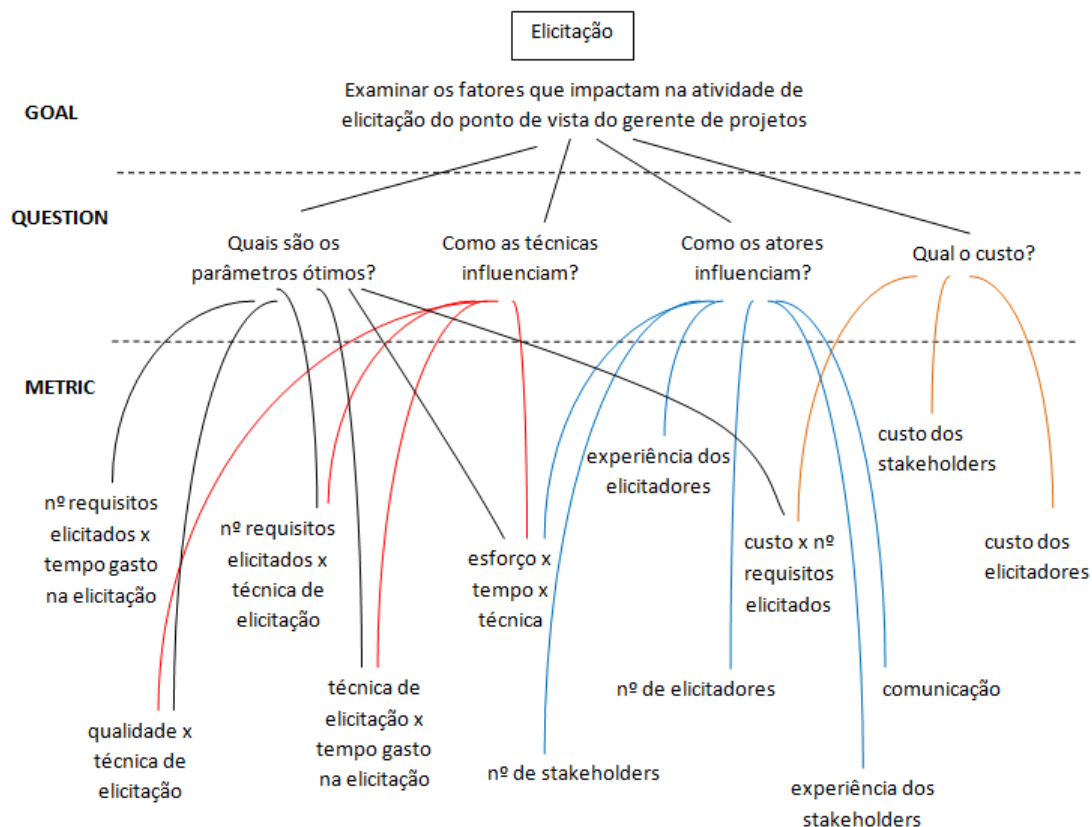
3.1 Identificação e seleção de variáveis

Tomando como ponto de partida a literatura citada no Capítulo 2 sobre engenharia de requisitos e elicitação de requisitos, aliada aos estudos (DIESTE; JURISTO, 2010; DIESTE; LOPEZ; RAMOS, 2008; DIESTE; JURISTO; SHULL, 2008; DAVIS *et al.*, 2006), foi possível analisar diversas variáveis presentes na atividade de elicitação.

Para auxiliar na identificação e seleção das principais variáveis dinâmicas da elicitação foi utilizado um método baseado no framework *Goal-Question-Metric* (GQM) (BASILI, 1992), que pode ser muito útil na modelagem de dinâmica de sistemas (MADACHY, 2007). O GQM possui três etapas principais: gerar um conjunto de objetivos organizacionais; derivar um conjunto de questões relacionadas aos objetivos; e desenvolver um conjunto de métricas necessárias para responder às questões (MADACHY, 2007).

A Figura 3.2 mostra um resumo da utilização do GQM neste trabalho. Nessa figura foram identificadas várias métricas que deram origem a variáveis dinâmicas da atividade de elicitação. Além da identificação das variáveis, as métricas também ajudaram a identificar algumas relações que deveriam ser feitas para que questões gerenciais pudessem ser respondidas.

As questões abordadas no GQM foram elaboradas baseando-se em questões importantes do ponto de vista gerencial abordados na literatura e validados por gerentes de software de empresas de Viçosa/MG.



Fonte: Elaborada pelo autor.

Figura 3.2. Exemplo de GQM utilizado no trabalho²

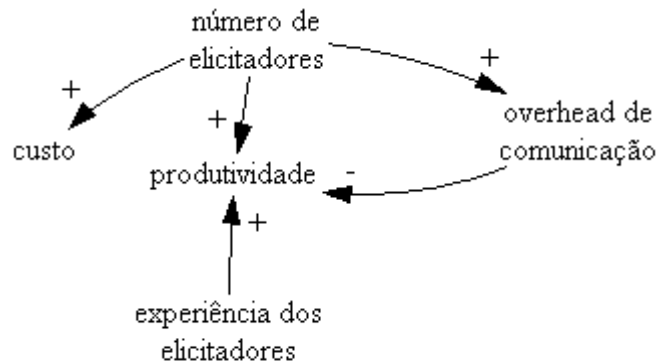
3.2 Diagramas de influência

Para entender como as principais variáveis dinâmicas que influenciam na atividade de elicitação se relacionam, foram utilizados diagramas de influência. Cada diagrama será explicado e exibido a seguir. Ao final dessa seção, será mostrado um diagrama completo, contemplando todas as partes explicadas anteriormente.

As relações dos elicitadores são mostradas no diagrama de influência da Figura 3.3. Quanto mais elicitadores estiverem alocados para o projeto, maior será a produtividade percebida, visto que mais pessoas farão o trabalho ao mesmo tempo. Porém, o custo será maior e maior também será o *overhead*³ de comunicação, que por sua vez acaba causando uma baixa na produtividade. Quanto maior a experiência dos elicitadores, maior será a produtividade percebida. A experiência dos elicitadores que foi levada em consideração neste trabalho está relacionada ao seu perfil, ou seja,

² O “x” utilizado no GQM tem o significado de “versus”. Por exemplo, “esforço x tempo” significa “esforço versus tempo”.

que tipo de elicitor está sendo utilizado no projeto (estudante, especialista no assunto ou engenheiro de software/analista) (GRIFFIN; HAUSER, 1993).



Fonte: Elaborada pelo autor.

Figura 3.3. Relações dos elicidores

As relações dos *stakeholders* são exibidas na Figura 3.4. Quanto maior o número de *stakeholders* no projeto, também será percebido um aumento na produtividade, visto que os elicidores poderão trabalhar em paralelo com os *stakeholders* e mais requisitos serão elicidados em menos tempo. Porém, aumentando-se o número de *stakeholders* também aumenta o custo do projeto, o *overhead* de comunicação e o tempo gasto na atividade. O acréscimo no tempo gasto na elicitação ocorre por aumentar o “trabalho” na atividade, ou seja, quanto mais *stakeholders* envolvidos, mais horas de elicitação serão necessárias para elicitar os requisitos de cada um dos *stakeholders*. Além disso, quanto mais *stakeholders* alocados maiores são as chances de elicitar requisitos repetidos, visto que um *stakeholder* não sabe exatamente o que já foi elicitado a partir dos outros. Quanto mais experientes são os *stakeholders*, ou seja, quanto maior os seus conhecimentos no domínio, maior a produtividade percebida (GRIFFIN; HAUSER, 1993; FOWLKES et al., 2000).

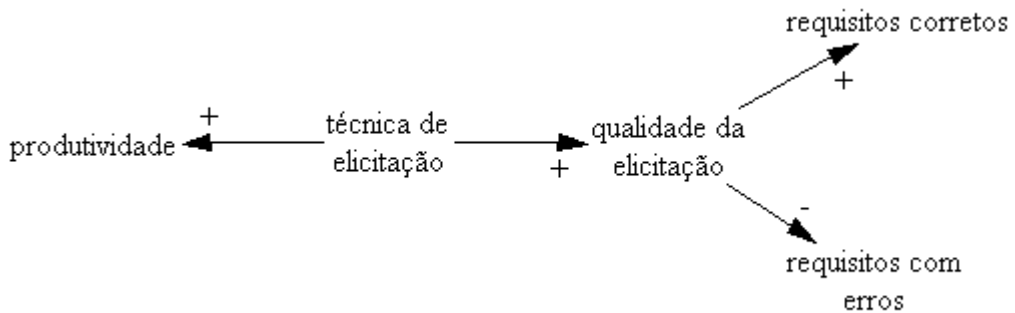
³ *Overhead* de comunicação aqui considerado como o tempo adicional gasto para os profissionais se comunicarem, que se torna maior à medida que o número de profissionais aumenta.



Fonte: Elaborada pelo autor.

Figura 3.4. Relações dos *stakeholders*

Uma variável chave para o modelo apresentado é a técnica de elicitação. Suas relações são exibidas no diagrama da Figura 3.5. A técnica utilizada está relacionada à produtividade e à qualidade da elicitação. Quanto melhor a técnica, maior será a produtividade percebida e maior será a qualidade da elicitação, ou seja, serão elicitados mais requisitos corretos e menos requisitos com erros (DIESTE; JURISTO, 2010).

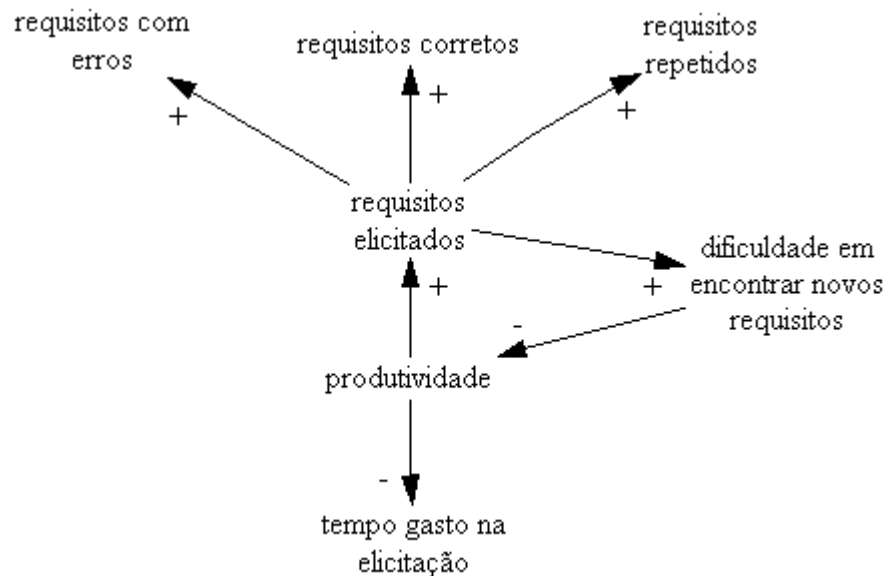


Fonte: Elaborada pelo autor.

Figura 3.5. Relações da técnica de elicitação

As relações da produtividade são mostradas na Figura 3.6. Quanto maior a produtividade, mais requisitos (corretos, errados ou repetidos) serão elicitados em menos tempo. Quanto mais requisitos são elicitados, mais difícil fica para serem encontrados novos requisitos. Quanto maior a dificuldade para elicitar novos requisitos, maior é a queda percebida na produtividade. Ou seja, mesmo mantendo

um mesmo cenário (mesmo número de pessoas alocadas, utilizando a mesma técnica), a tendência é que com o passar do tempo a produtividade caia na medida em que os requisitos vão sendo elicitados (GRIFFIN; HAUSER, 1993).



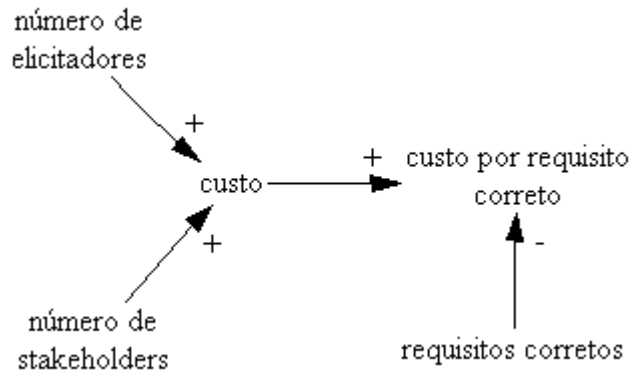
Fonte: Elaborada pelo autor.

Figura 3.6. Relações da produtividade

As relações dos custos do projeto são mostradas na Figura 3.7. Quanto mais profissionais alocados para a elicitação, maior é o custo geral do projeto. Geral no sentido de estar relacionado ao próprio projeto, independente de quem está arcando com esses custos, se é a empresa que está desenvolvendo o software ou a empresa contratante. A empresa de software é responsável pelos honorários dos elicidores, enquanto a empresa contratante arca com os custos dos *stakeholders*.

À primeira vista pode parecer que a relação dos custos dos *stakeholders* não é importante para o gerente de projetos da empresa de software (usuário do modelo); porém, a alocação de um *stakeholder* no projeto implica na parada de suas atividades normais dentro da empresa contratante. Se muitos *stakeholders* forem alocados, ou muitas horas de elicitação forem necessárias, isso pode gerar um custo muito grande para a empresa contratante, o que pode até inviabilizar o projeto. Além disso, existe um fator não considerado no modelo (pela dificuldade de mensuração), mas visto na prática, que é a resistência dos próprios *stakeholders* em auxiliar na elicitação. Quanto mais tempo o *stakeholder* é afastado das suas atividades normais, mais aumenta a sua resistência no auxílio da elicitação do projeto de software, visto que a preocupação com o acúmulo de suas atividades normais também aumenta.

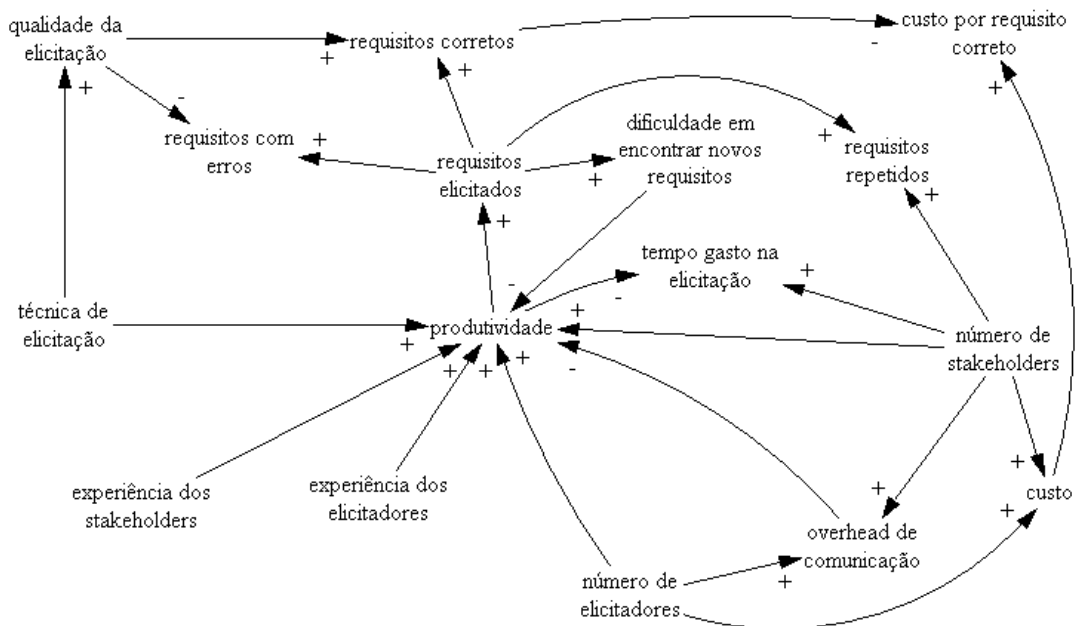
Para facilitar a comparação de custos entre os cenários simulados, foi calculado o custo por requisito correto, ou seja, o custo total do projeto dividido pelo número de requisitos corretos elicidados.



Fonte: Elaborada pelo autor.

Figura 3.7. Relações de custos

Um diagrama de influência contemplando toda a atividade de elicitação é exibido na Figura 3.8. Todas as relações detalhadas nesta seção estão presentes nesse diagrama.



Fonte: Elaborada pelo autor.

Figura 3.8. Diagrama de influência da atividade de elicitação

3.3 Diagrama estoque-fluxo

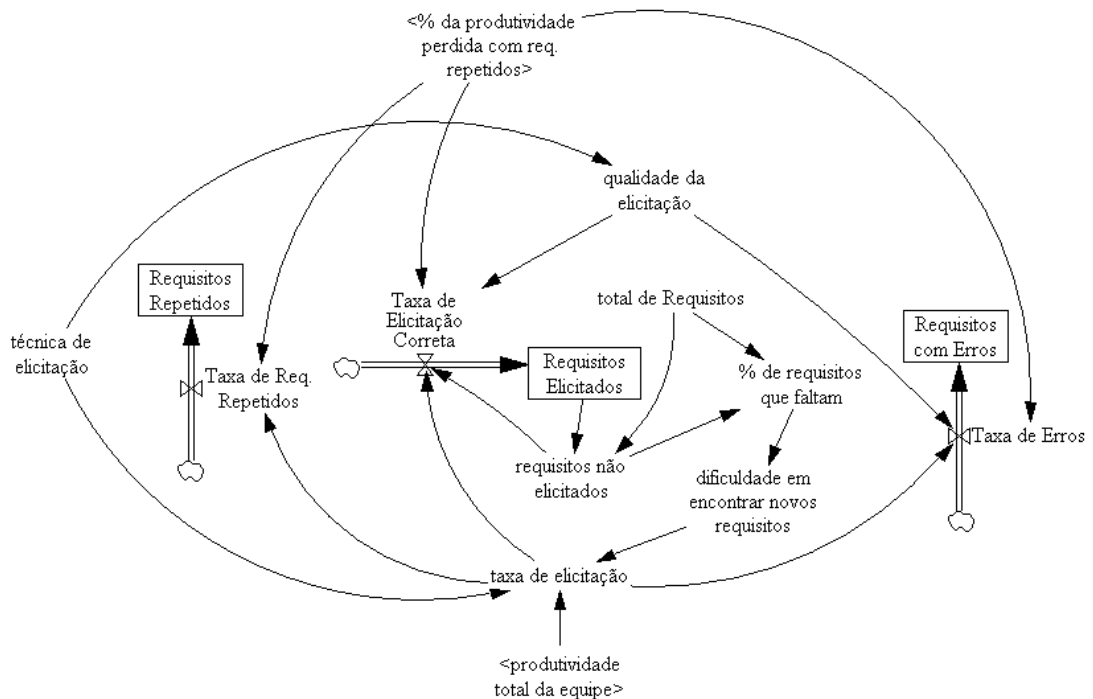
A partir das relações estabelecidas nos diagramas de influência apresentados na seção anterior, foi construído um diagrama estoque-fluxo da dinâmica de sistemas para representar a atividade de elicitação. A transformação do diagrama de influência no diagrama estoque-fluxo foi baseada no processo disponível em (MADACHY, 2007).

Para facilitar o entendimento, cada parte do diagrama estoque-fluxo será apresentada separadamente, com as explicações de como as variáveis foram definidas e quantificadas.

3.3.1 Taxa de Elicitação

A primeira parte do modelo (Figura 3.9) apresenta o fluxo principal, ou seja, a taxa de elicitação correta, taxa de requisitos repetidos e a taxa de erros produzidos na elicitação.

Essa parte do modelo é influenciada pela técnica de elicitação utilizada e pela produtividade da equipe.



Fonte: Elaborada pelo autor.

Figura 3.9. Fluxo principal do modelo

A “taxa de elicitação” depende da “técnica de elicitação” utilizada e da “produtividade total da equipe” alocada no projeto. Essa taxa influencia a quantidade de requisitos elicitados. Os requisitos elicitados podem ser requisitos repetidos ou novos requisitos, que ainda podem ter sido elicitados corretamente (“Taxa de Elicitação Correta”) ou não (“Taxa de Erros”), dependendo da “qualidade da elicitação”. A “qualidade da elicitação”, por sua vez, depende também da “técnica de elicitação” utilizada.

Baseado no trabalho (DIESTE; JURISTO, 2010), foi possível construir diversas tabelas de comparação entre as técnicas de elicitação considerando vários quesitos: eficiência (tempo), eficácia (quantidade), completude, desempenho e qualidade. O resultado final no que diz respeito à efetividade das técnicas é apresentado na Tabela 3.1. Nessa tabela, o sinal “>” indica que a técnica da linha se mostrou mais efetiva do que a técnica da coluna correspondente ao sinal. O sinal “<” significa que a técnica apresentada na linha se mostrou menos efetiva que a técnica mostrada na coluna correspondente ao sinal. Por exemplo, a técnica de entrevista se mostrou mais efetiva do que a técnica de introspecção e observação. Já a técnica de análise de protocolo foi considerada menos efetiva que a técnica de *sorting*.

Tabela 3.1. Comparação entre as técnicas de elicitação baseada em (DIESTE; JURISTO, 2010)

Efetividade			
	Introspecção e observação	<i>Sorting</i>	<i>Laddering</i>
Entrevista	>	>	>
Análise de Protocolo		<	<
<i>Sorting</i>			<

Fonte: Elaborada pelo autor

Com base na tabela de comparação, é possível perceber que a ordem das técnicas (da mais para a menos efetiva) é: entrevista, *laddering*, *sorting* e por fim análise de protocolo (introspecção e observação). Essa generalização proposta pôde ser validada através de diferentes trabalhos.

No próprio trabalho tomado como base para a generalização (DIESTE; JURISTO, 2010) é afirmado que entrevistas são equivalentes ou mais efetivas do que análise de protocolo e *sorting*. Entrevistas conseguem obter requisitos mais

completos do que análise de protocolo, *sorting* e *laddering*. É dito ainda que a análise de protocolo é a pior de todas as técnicas em todas as dimensões (efetividade, eficiência e completude).

DIESTE; LOPEZ; RAMOS (2008) afirmam que em suas pesquisas a entrevista foi a técnica de elicitación mais efetiva. Em geral, a entrevista apresentou melhor desempenho do que *sorting* e análise de protocolo, com muito poucas exceções. *Laddering* é quase tão boa quanto a entrevista. *Sorting* e análise de protocolo podem ser consideradas como técnicas de baixa efetividade. Por fim, afirmam que, em geral, *sorting* é melhor do que análise de protocolo.

No trabalho (DAVIS *et al.*, 2006), os resultados de agregações mostram que entrevistas obtêm mais informações do que *sorting* e análise de protocolo.

Considerando todos essas afirmações dos trabalhos citados, pode-se considerar como válida a generalização realizada neste trabalho. Porém, até agora foi mostrado somente uma comparação qualitativa entre as técnicas. Para a execução do modelo de dinâmica de sistemas, deve-se estabelecer uma relação quantitativa. Assim, foram buscadas outras referências de experimentos na literatura onde as técnicas foram comparadas quantitativamente.

CORBRIDGE (*et al.*, 1994) e BURTON (*et al.*, 1990) realizaram experimentos de comparação entre quatro técnicas de elicitación: entrevista, *laddering*, *sorting* e análise de protocolo. Nesses experimentos, as sessões de elicitación foram gravadas e depois codificadas em “*Pseudo-English Production Rules* (PPRs)”. Essas regras (*rules*) são compostas por estruturas *IF*, *AND* e *THEN*, mas são independentes de implementação em software (BURTON *et al.*, 1990). Regras como essas são úteis para avaliar o “ganho” da técnica, ou seja, a quantidade de conhecimento que é elicitada a partir dos *stakeholders* utilizando uma técnica específica (CORBRIDGE *et al.*, 1994). Porém, essa medida de ganho não reflete precisamente o número de informações contidas nas regras (*rules*), uma vez que uma regra pode conter muitas cláusulas (CORBRIDGE *et al.*, 1994). Uma cláusula é definida como uma instrução condicional em uma regra de produção (PPR) (BURTON *et al.*, 1990). Por exemplo, seja uma regra “*IF a AND b AND c THEN x*” e outra regra “*IF d THEN y*”. Na primeira regra existem quatro cláusulas (“*IF a*”, “*AND b*”, “*AND c*” e “*THEN x*”), enquanto na segunda existe apenas duas (“*IF d*” e

"THEN y"). Apesar de ambas representarem apenas uma regra de produção, é possível extrair mais informações da primeira regra, visto que ela contém mais cláusulas. Nos experimentos citados foram considerados os números de cláusulas para a comparação entre as técnicas de elicitação.

Os estudos (CORBRIDGE *et al.*, 1994; BURTON *et al.*, 1990; BURTON *et al.*, 1987; BURTON *et al.*, 1988) foram levados em consideração no presente trabalho por terem sido utilizados e bem avaliados em um dos trabalhos tomados como base (DIESTE; JURISTO, 2010). Como esses estudos compararam as técnicas em termos de número de cláusulas, essa também foi a medida adotada neste trabalho.

A Tabela 3.2 mostra uma compilação de resultados de comparações entre as técnicas de elicitação. Os dados omitidos na tabela não foram divulgados nos trabalhos citados. Os campos da tabela se referem ao número de cláusulas obtidas, tempo total da elicitação em minutos, a completude dos requisitos em relação ao total (em porcentagem) para um *stakeholder* e somando as cláusulas de todos os *stakeholders* (32 ao todo). Os *stakeholders* eram estudantes com conhecimentos sobre uma determinada área, e a porcentagem de completude alcançada por eles foi comparada à quantidade de cláusulas obtidas por um *expert* sênior na área (considerado 100%).

A relação da “técnica de elicitação” com a “taxa de elicitação” no modelo (mostrado na Figura 3.9) foi considerada baseando-se nas médias de cláusulas por tempo obtidas na Tabela 3.2. Uma observação importante é que essas médias foram baseadas a partir de *stakeholders* pouco experientes (estudantes). Os dados divulgados pelo Exp. 5 foram muito superiores quando comparados aos divulgados nos experimentos anteriores, portanto esses resultados não foram considerados nos cálculos das médias. Os valores elevados podem ter sido consequência do domínio do problema em questão (diagnóstico médico) e da forma como o experimento foi conduzido, que foi diferenciado na comparação com os demais.

Tabela 3.2. Resumo dos resultados dos experimentos descritos em (CORBRIDGE et al., 1994; BURTON et al., 1990)

Técnica	Medida	Exp 1⁴	Exp 2⁵	Exp 3⁶	Exp 4⁷	Exp 5⁸
<i>Laddering</i>	Cláusulas	101,4	--	188	278	521,4
	Tempo (min)	79,8	--	145	177	40,75
	Compleitude % (1)	30	35,3	--	--	--
	Compleitude % (32)	85,2	100	--	--	--
	Cláusulas/Tempo	1,3	--	1,3	1,6	12,8
<i>Sorting</i>	Cláusulas	63,4	--	123	216	144
	Tempo (min)	67	--	98	145	29,75
	Compleitude % (1)	30	23,4	--	--	--
	Compleitude % (32)	81,8	86	--	--	--
	Cláusulas/Tempo	0,9	--	1,3	1,5	4,8
Entrevista	Cláusulas	94,4	--	270	317	274
	Tempo (min)	--	--	217	240	39,5
	Compleitude % (1)	27,9	33	--	--	--
	Compleitude % (32)	79,5	94	--	--	--
	Cláusulas/Tempo	--	--	1,2	1,3	6,9
Análise de Protocolo	Cláusulas	75,8	--	269	184	145
	Tempo (min)	--	--	351	176	26,75
	Compleitude % (1)	7,9	16	--	--	--
	Compleitude % (32)	56,8	66	--	--	--
	Cláusulas/Tempo	--	--	0,8	1,0	5,4

Fonte: Elaborada pelo autor

No modelo da Figura 3.9, a relação entre a “técnica de elicitação” e a “qualidade da elicitação” foi baseada no trabalho (BURTON et al., 1990). Um dos experimentos realizados nesse trabalho foi a classificação por parte dos *stakeholders* do conjunto de cláusulas elicitadas a partir deles mesmos pelos elicitadores. As

⁴ (BURTON et al., 1987 apud CORBRIDGE et al., 1994)

⁵ (BURTON et al., 1988 apud CORBRIDGE et al., 1994)

⁶ (BURTON et al., 1990)

⁷ (BURTON et al., 1990)

⁸ (CORBRIDGE et al., 1994)

cláusulas foram categorizadas em verdadeiras, falsas, triviais e parcialmente verdadeiras. As definições de verdadeiras e falsas são auto-explicativas. A definição de trivial dada foi que a cláusula não é relevante no diagnóstico de problemas no domínio em questão. Uma cláusula classificada como parcialmente verdadeira significa que se ela sofresse alguma modificação ela se tornaria verdadeira (BURTON et al., 1990). Os resultados do experimento são exibidos na Tabela 3.3.

Tabela 3.3. Resultados do experimento realizado em (BURTON et al., 1990)

	Verdadeira	Trivial	Parcialmente Verdadeira	Falsa
Entrevista	63	22	8	6
Análise de Protocolo	46	17	32	5
<i>Sorting</i>	43	44	7	6
<i>Laddering</i>	55	22	17	6

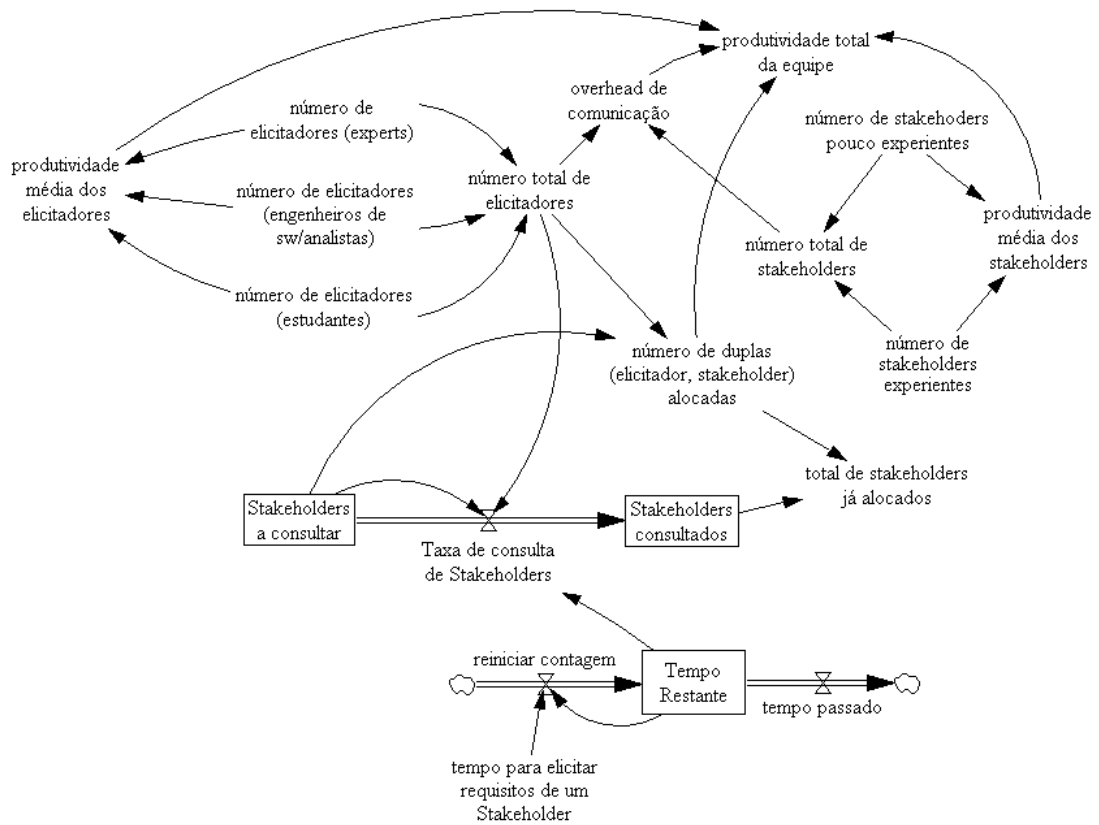
Fonte: Traduzida de (BURTON et al., 1990)

Para quantificar a qualidade da elicitación foram levados em consideração os resultados mostrados na Tabela 3.3. Foram consideradas como cláusulas elicítadas corretamente as cláusulas classificadas como verdadeiras e triviais, porém, foi dado um peso maior para as cláusulas classificadas como verdadeiras, já que são as cláusulas mais úteis no contexto do problema que o software deve resolver.

Para finalizar essa parte do modelo (Figura 3.9), foi considerada a dificuldade em encontrar novos requisitos. Essa dificuldade foi estimada com base em entrevistas feitas com engenheiros de softwares de uma empresa de Viçosa-MG. A dificuldade é inversamente proporcional ao número de cláusulas que restam para ser elicítadas no domínio do problema. Quanto menos cláusulas restam, maior é a dificuldade em encontrá-las e elicítá-las. Além disso, a relação não é linear. A dificuldade aumenta de forma exponencial na medida em que as cláusulas vão sendo elicítadas.

3.3.2 Produtividade

A segunda parte do modelo (Figura 3.10) apresenta as variáveis que impactam na produtividade da equipe.



Fonte: Elaborada pelo autor.

Figura 3.10. Parte do modelo referente à produtividade

A relação entre a produtividade dos elicitadores com o seu perfil (especialista, engenheiro ou estudante) foi quantificada baseando-se no experimento realizado em (GRIFFIN; HAUSER, 1993). Segundo esse experimento, em média cada elicitador foi capaz de identificar 54% das necessidades dos *stakeholders*, variando-se entre 45 a 68%, sendo que os especialistas ficaram na extremidade baixa (45%) e a equipe de engenheiros ficou na extremidade alta (68%). Os estudantes ficaram na metade do intervalo (considerado 56%). Para o modelo proposto neste trabalho, não foram considerados os valores conforme apareceram em (GRIFFIN; HAUSER, 1993), mas sim a diferença entre eles. Foi considerado o perfil de engenheiro como 100%, obtendo-se que os especialistas têm 66% da produtividade dos engenheiros, enquanto os estudantes têm 82%.

A diferença entre a produtividade dos *stakeholders* experientes e pouco experientes, ou seja, com muito ou pouco conhecimento no domínio, foi baseada na Tabela 3.2. Nela é mostrada a completude alcançada por um estudante (menos experiente) com relação a um especialista sênior da área (muito experiente). O fato

de os *stakeholders* mais experientes obterem maior produtividade do que os menos experientes pode ser validado através do trabalho (FOWLKES *et al.*, 2000).

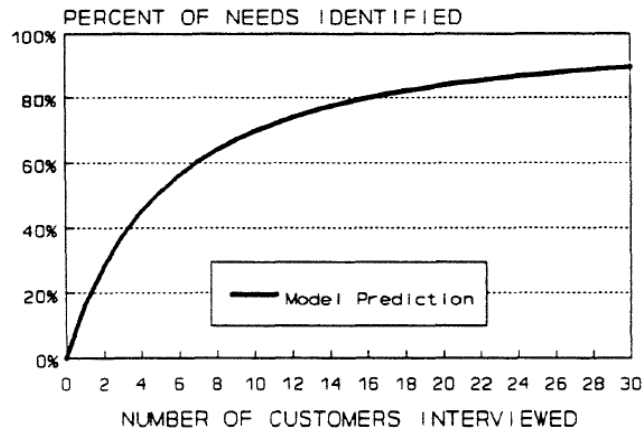
Apesar do acréscimo do número de elicitadores e *stakeholders* no projeto poder aumentar a produtividade percebida, quanto maior for o tamanho da equipe, maior será o tempo gasto com comunicação e reuniões envolvendo os membros da equipe (SIMMONS, 1991), ou seja, maior será o *overhead* de comunicação (MADACHY, 2007). Esse *overhead* é dado em porcentagem e expresso como uma função não-linear do total de pessoas que precisam se comunicar (ABDEL-HAMID; MADNICK, 1991).

No desenvolvimento do modelo, foi considerado que os elicitadores e *stakeholders* trabalham sempre em duplas (um elicitador com um *stakeholder*), até que todos os *stakeholders* sejam consultados. Dessa forma, o número de *stakeholders* inicial é dividido pelo número de elicitadores e essa “sessão de elicitação” tem um tempo estipulado. Passado o tempo de elicitação, os elicitadores começam uma nova sessão com os *stakeholders* que ainda não foram consultados.

No modelo da Figura 3.10, a “produtividade total da equipe” em um dado momento pode ser expressa como uma função do número de duplas (elicitador e *stakeholder*) alocadas no momento multiplicado pelas produtividades médias dos *stakeholders* e elicitadores, descontando-se a produtividade perdida pelo *overhead* de comunicação.

3.3.3 Requisitos repetidos

Além de a produtividade do *stakeholder* estar relacionada com a sua experiência, existe uma relação não linear entre o número de *stakeholders* alocados no projeto e o percentual de necessidades identificadas. Essa relação está confirmada por estudos realizados em (GRIFFIN; HAUSER, 1993), e pode ser visualizada no gráfico exibido na Figura 3.11.



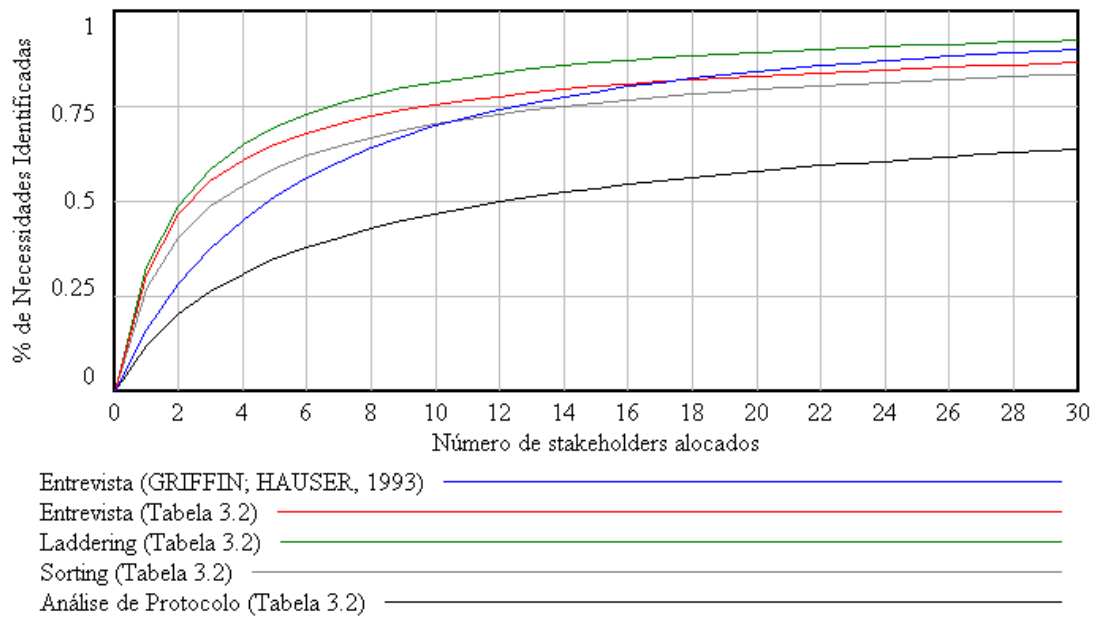
Fonte: (GRIFFIN; HAUSER, 1993)

Figura 3.11. Relação entre número de clientes entrevistados e o percentual de necessidades identificadas

A relação exibida na Figura 3.11 pode ser obtida através de um modelo beta-binomial (GRIFFIN; HAUSER, 1993). A partir desse modelo é possível prever de forma aproximada qual é a produtividade percebida (através do percentual de necessidades identificadas) dado o número de pessoas entrevistadas. Essa relação não é linear devido ao número de requisitos repetidos elicitados pelos *stakeholders*, que aumenta na medida em que o número de *stakeholders* também aumenta.

Em vez de utilizar os valores disponíveis em (GRIFFIN; HAUSER, 1993) e exibidos na Figura 3.11, foi utilizado apenas o modelo beta-binomial para aplicar aos dados das técnicas de elicitación obtidos através da Tabela 3.2.

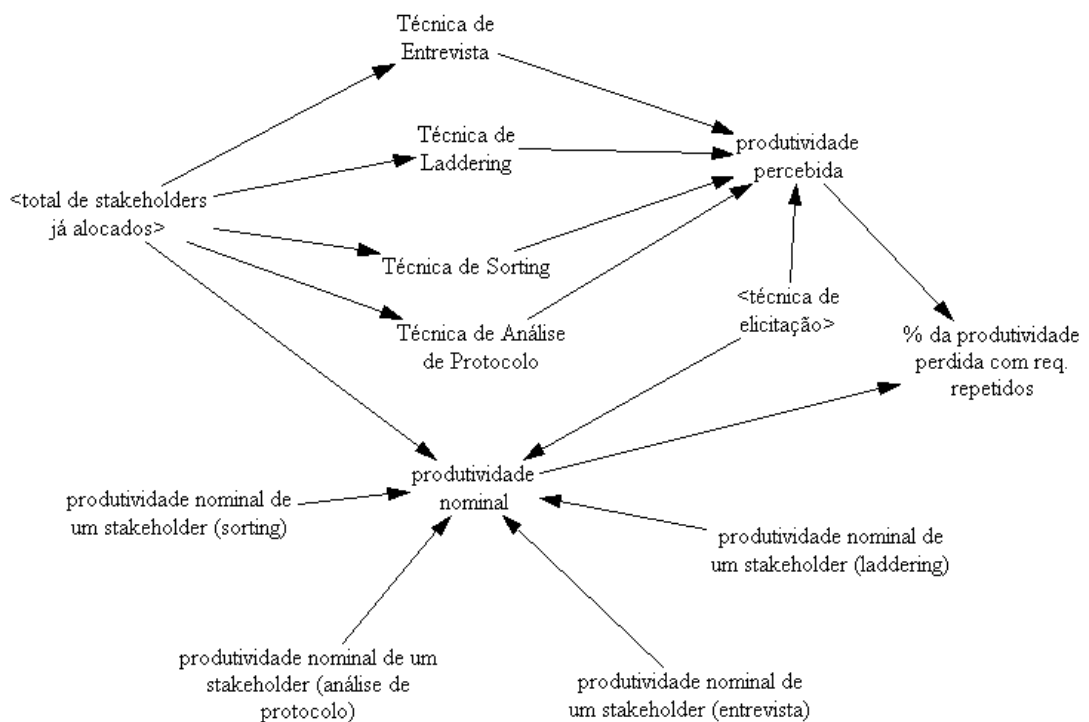
A partir do modelo beta-binomial (GRIFFIN; HAUSER, 1993) foi possível traçar cinco curvas (uma para cada técnica e duas para a técnica de entrevista – uma baseada em (Griffin; Hauser, 1993) e outra baseada nos dados da Tabela 3.2). Essas curvas são exibidas na Figura 3.12 e foram utilizadas para calcular o quanto de produtividade é perdido com requisitos repetidos dado o número de *stakeholders* alocados.



Fonte: Elaborada pelo autor

Figura 3.12. Comparação entre os gráficos obtidos utilizando a função beta-binomial

A parte do modelo que descreve a modelagem dos requisitos repetidos é exibida na Figura 3.13.



Fonte: Elaborada pelo autor

Figura 3.13. Parte do modelo referente a requisitos repetidos

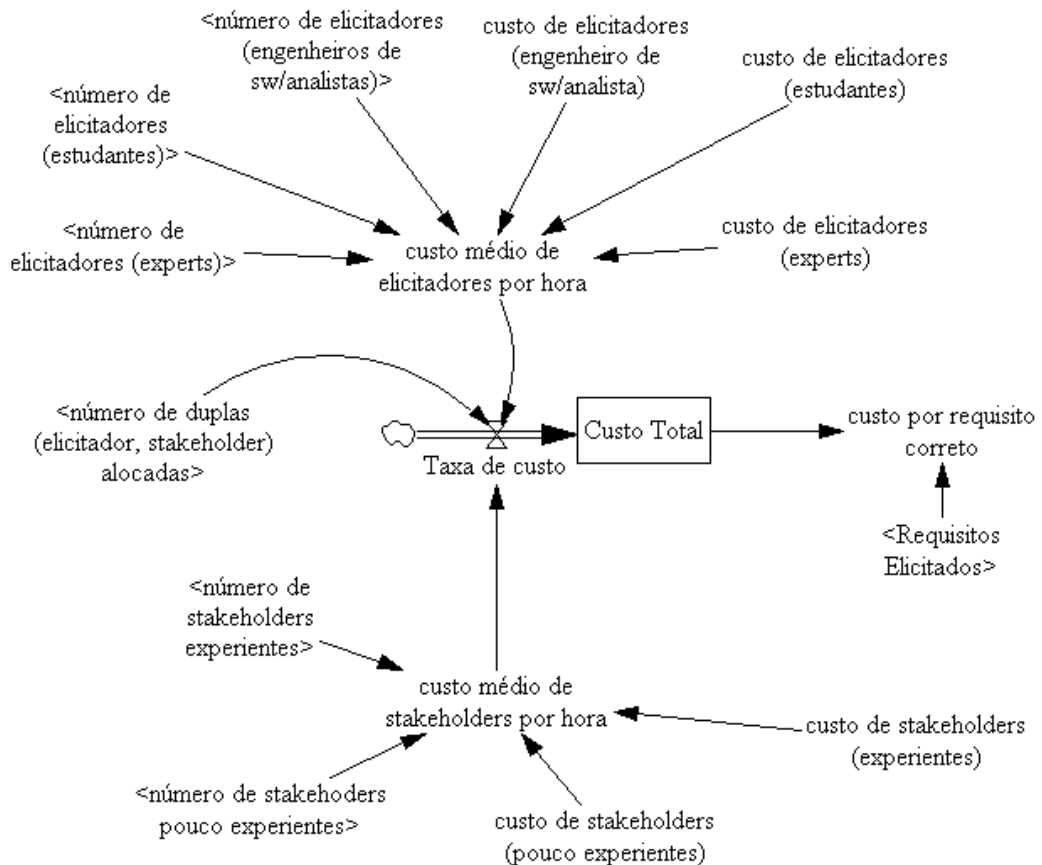
A “porcentagem da produtividade perdida com requisitos repetidos” é calculada a partir da diferença entre a produtividade nominal de um *stakeholder* e a

produtividade média percebida dos *stakeholders* alocados no projeto. A “produtividade nominal” é a produtividade de um único *stakeholder* alocado no projeto utilizando uma técnica de elicitação específica. A “produtividade percebida” é a média das produtividades dos *stakeholders* alocados.

3.3.4 Custo da elicitação

A quarta parte do modelo a ser analisada mostra os custos envolvidos na elicitação. Para o propósito deste trabalho, é necessário apenas que exista um parâmetro de comparação de custos entre os diversos cenários a serem simulados a partir do modelo proposto. Portanto, não é necessário que esses custos sejam representados por valores exatos.

A parte do modelo que trata dos custos é apresentada na Figura 3.14.



Fonte: Elaborada pelo autor

Figura 3.14. Parte do modelo referente aos custos

Nessa parte do modelo são calculados os custos médios dos profissionais na equipe dado o número de pessoas de cada perfil multiplicado pelo valor da remuneração desse perfil.

Além disso, é calculado o custo médio por requisito correto, dividindo o custo total pelo número total de requisitos elicitados corretamente até o momento.

4 SIMULAÇÕES E ANÁLISES DOS RESULTADOS

Para executar simulações a partir do modelo apresentado na seção anterior, foram criados vários cenários que abrangem diferentes configurações de fases iniciais de projetos de software. Esses cenários facilitam as análises e comparações entre as simulações.

Em todos os cenários simulados, os custos por hora dos profissionais foram fixados em: 20,00UM (unidades monetárias) por elicitor engenheiro/analista, 5,00UM por elicitor estudante, 40,00UM por *stakeholder* experiente e 15,00UM por *stakeholder* pouco experiente. Além disso, foi considerada a simulação de 30 horas em sessões de elicitação, que foram divididas de acordo com o número de profissionais alocados. Foi também considerado um total de cinco mil cláusulas para serem elicitadas no projeto, o que irá interferir somente na dificuldade em elicitar novos requisitos.

4.1 Técnica de Entrevista

Como a técnica de entrevista é a mais utilizada atualmente, foi dedicada uma seção para comparar cenários utilizando somente essa técnica.

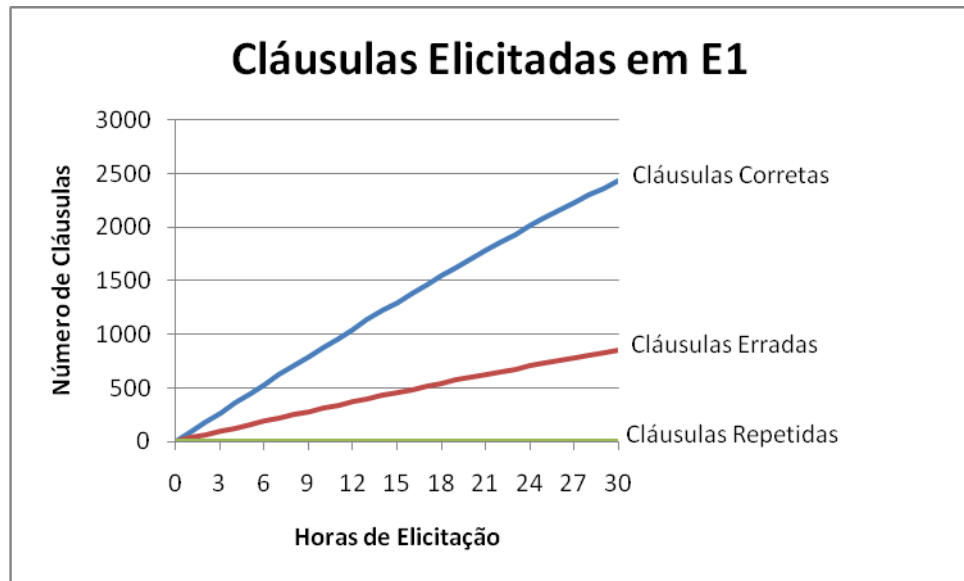
Os cenários foram escolhidos de forma a variar do cenário mais simples possível (E1), até cenários com o dobro de profissionais de E1 (E2 e E3). Os cenários E4 e E5 são intermediários. Essa distribuição facilitou a comparação entre os cenários.

A seguir, serão fornecidos os resultados da simulação de cada cenário e então será feita uma comparação entre eles.

4.1.1 Cenário E1

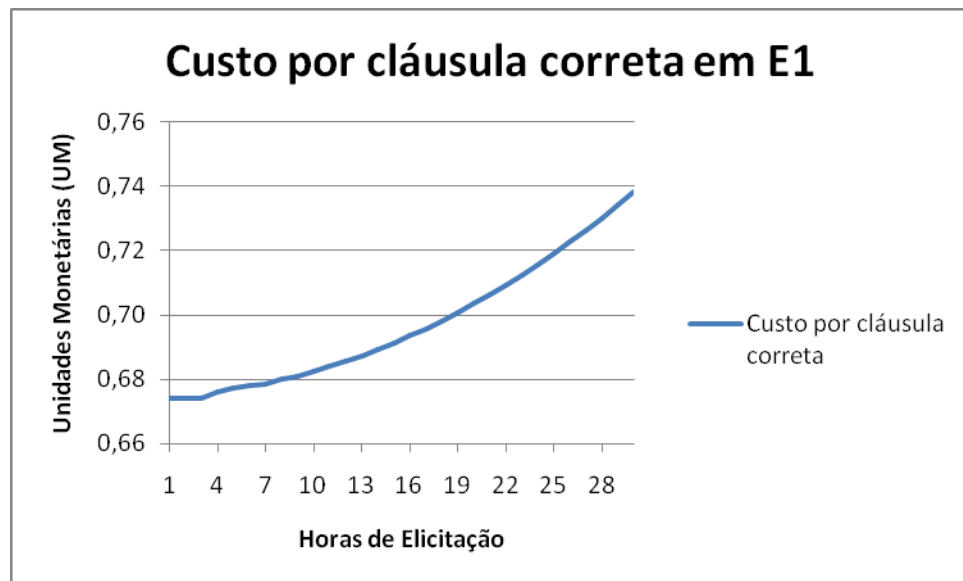
O cenário E1 é composto por apenas 1 elicitor engenheiro de software/analista e 1 *stakeholder* experiente.

Os resultados referentes às cláusulas elicítadas são exibidos na Figura 4.1. O gráfico do custo por requisito correto é mostrado na Figura 4.2.



Fonte: Elaborada pelo autor

Figura 4.1. Cláusulas elicítadas no Cenário E1



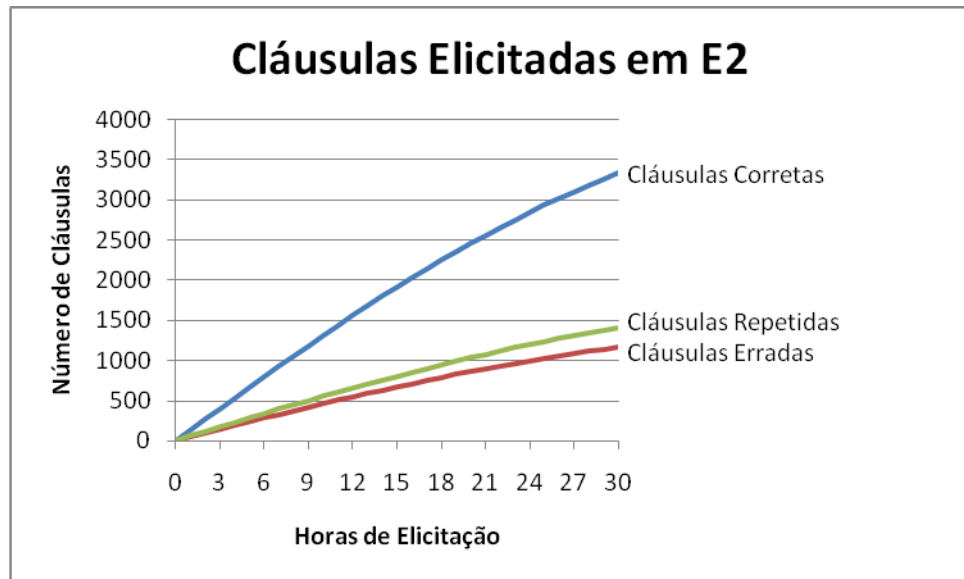
Fonte: Elaborada pelo autor

Figura 4.2. Custo por cláusula correta no Cenário E1

4.1.2 Cenário E2

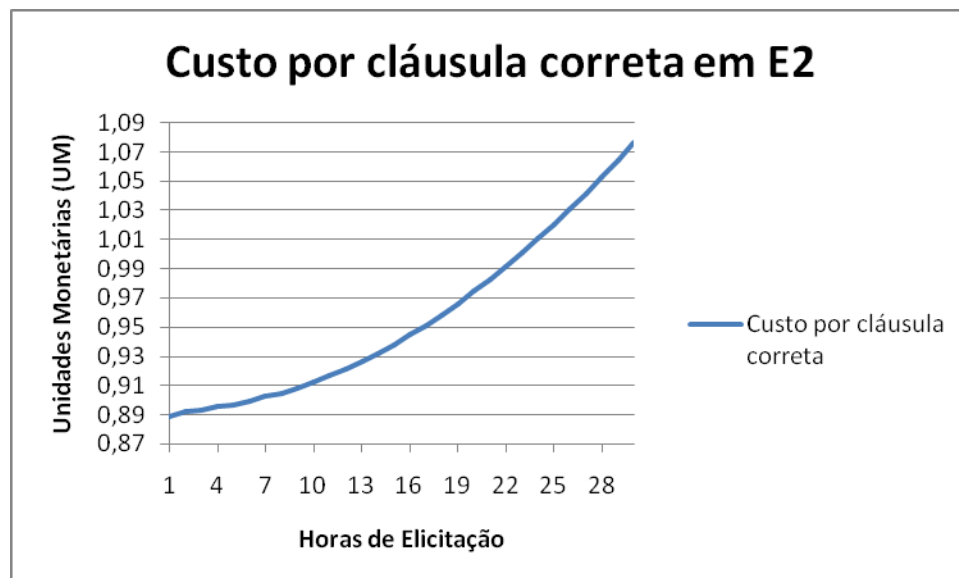
O cenário E2 é formado por 2 elicítadores engenheiros de software/analistas e 2 *stakeholders* experientes, ou seja, o dobro de profissionais com relação a E1.

Os resultados da simulação são apresentados na Figura 4.3 e Figura 4.4.



Fonte: Elaborada pelo autor

Figura 4.3. Cláusulas elicítadas no Cenário E2



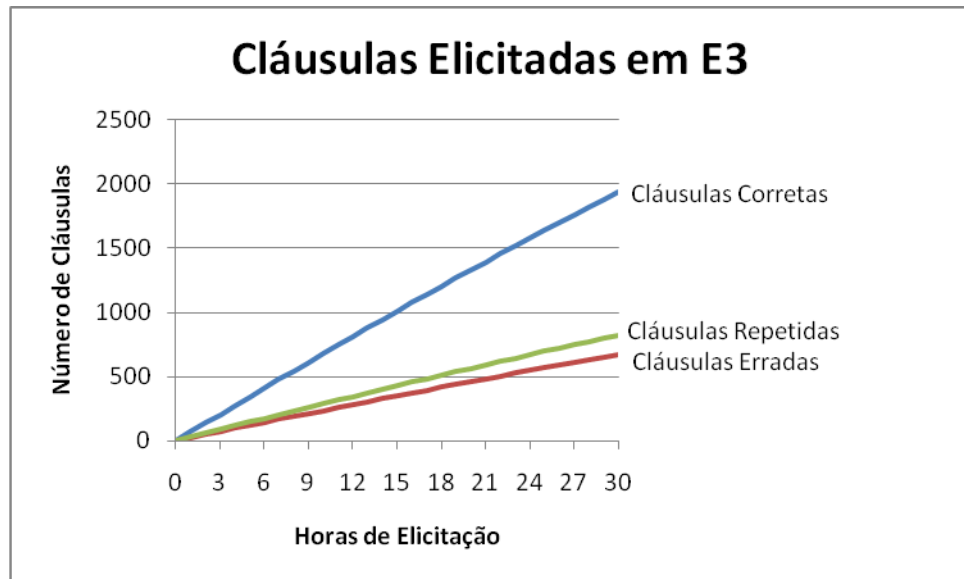
Fonte: Elaborada pelo autor

Figura 4.4. Custo por cláusula correta no Cenário E2

4.1.3 Cenário E3

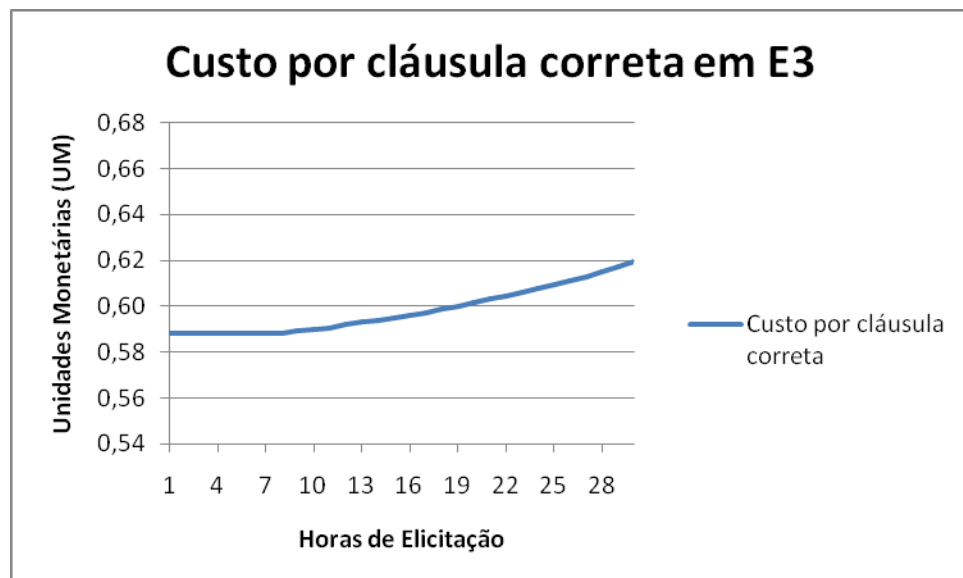
No cenário E3 são considerados profissionais com menos experiência, uma mão de obra mais barata. Nesse cenário, são alocados 2 elicítadores estudantes e 2 *stakeholders* pouco experientes.

Os resultados da simulação são apresentados na Figura 4.5 e Figura 4.6.



Fonte: Elaborada pelo autor

Figura 4.5. Cláusulas elicítadas no Cenário E3



Fonte: Elaborada pelo autor

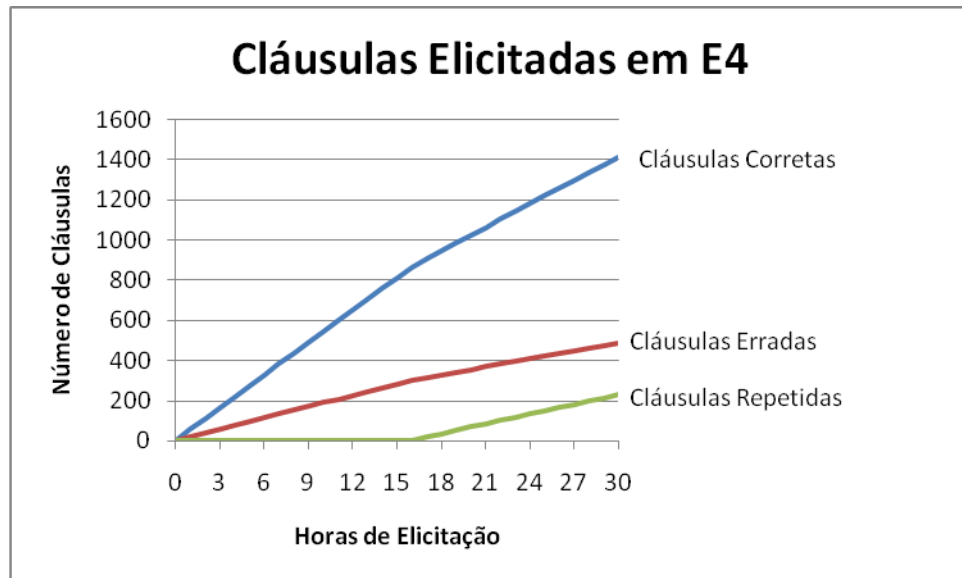
Figura 4.6. Custo por cláusula correta no Cenário E3

4.1.4 Cenário E4

No cenário E4 são alocados 1 elicítador engenheiro de software/analista e 2 *stakeholders* pouco experientes. Neste caso, foram alocadas 15 horas de sessões de elicitação para cada *stakeholder*, totalizando as mesmas 30 horas em sessões de elicitação dos outros cenários.

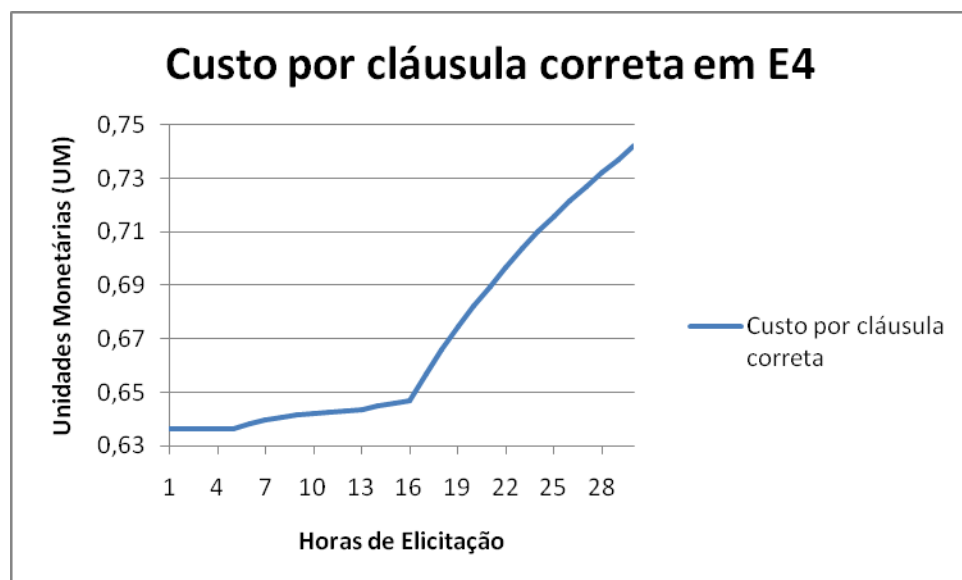
A Figura 4.7 e Figura 4.8 mostram os resultados da simulação. Na Figura 4.7, é interessante observar que a partir do momento que o elicítador termina de elicitar

os requisitos do primeiro *stakeholder* e começa a trabalhar com o segundo (em torno da hora 15 na simulação), começa a aumentar no gráfico o número de cláusulas repetidas. Como consequência, essas cláusulas repetidas diminuem a produtividade real da equipe (relacionada aos requisitos corretos), o que acaba causando um aumento do custo por cláusula correta, observado na Figura 4.8.



Fonte: Elaborada pelo autor

Figura 4.7. Cláusulas elicitadas no Cenário E4



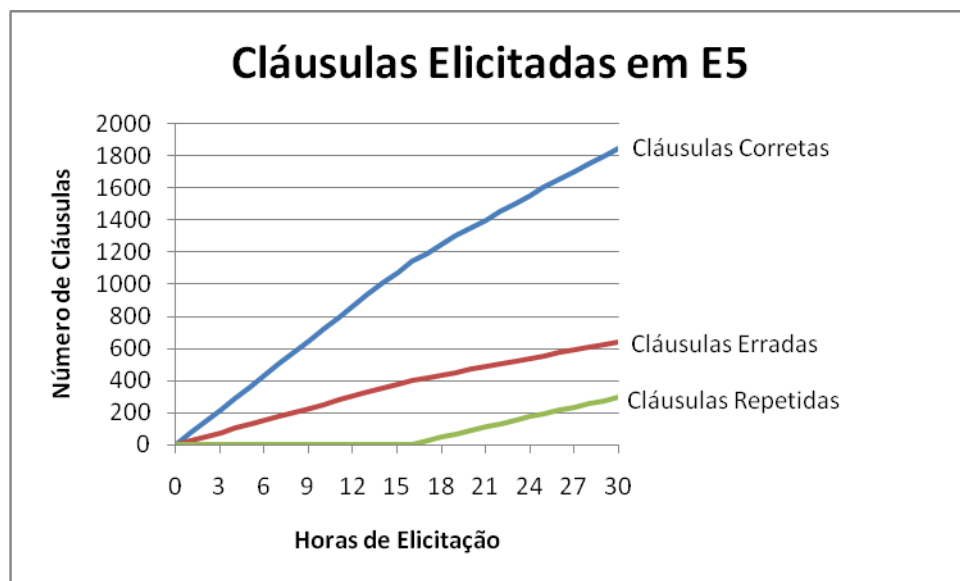
Fonte: Elaborada pelo autor

Figura 4.8. Custo por cláusula correta no Cenário E4

4.1.5 Cenário E5

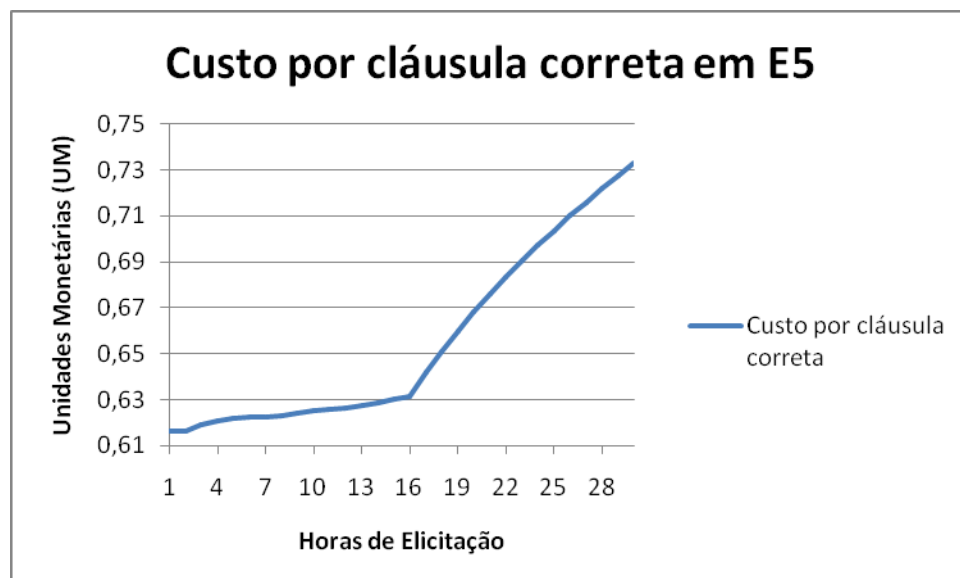
No cenário E5 são considerados 1 elicitador estudante e 2 *stakeholders* experientes. Assim como em E4, foram alocadas 15 horas de sessões de elicitação para cada *stakeholder*, totalizando 30 horas em sessões de elicitação.

Os resultados da simulação são exibidos na Figura 4.9 e Figura 4.10. Assim como em E4, a partir da hora 15 na simulação é observado um aumento do número de cláusulas repetidas, que causam um aumento do custo por cláusula correta.



Fonte: Elaborada pelo autor

Figura 4.9. Cláusulas elicítadas no Cenário E5



Fonte: Elaborada pelo autor

Figura 4.10. Custo por cláusula correta no Cenário E5

4.1.6 Comparação dos cenários utilizando a técnica de entrevista

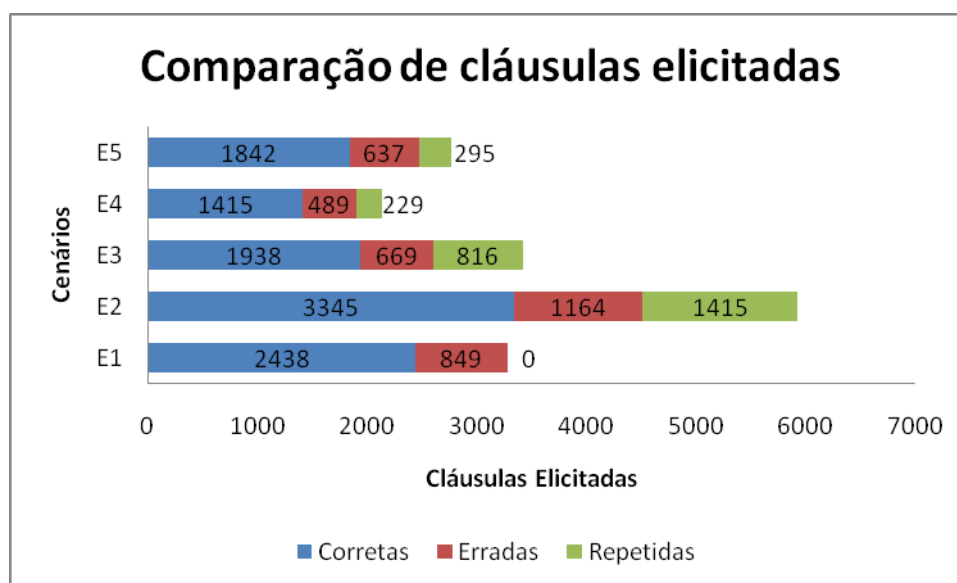
Para facilitar a visualização da configuração dos cenários, eles foram resumidos na Tabela 4.1.

Tabela 4.1. Resumo dos cenários de entrevista

Cenário E1	Cenário E2	Cenário E3	Cenário E4	Cenário E5
1 elicitador engenheiro de software/analista	2 elicitadores engenheiros de software/analistas	2 elicitadores estudantes	1 elicitador engenheiro de software/analista	1 elicitador estudante
1 <i>stakeholder</i> experiente	2 <i>stakeholders</i> experientes	2 <i>stakeholders</i> pouco experientes	2 <i>stakeholders</i> pouco experientes	2 <i>stakeholders</i> experientes

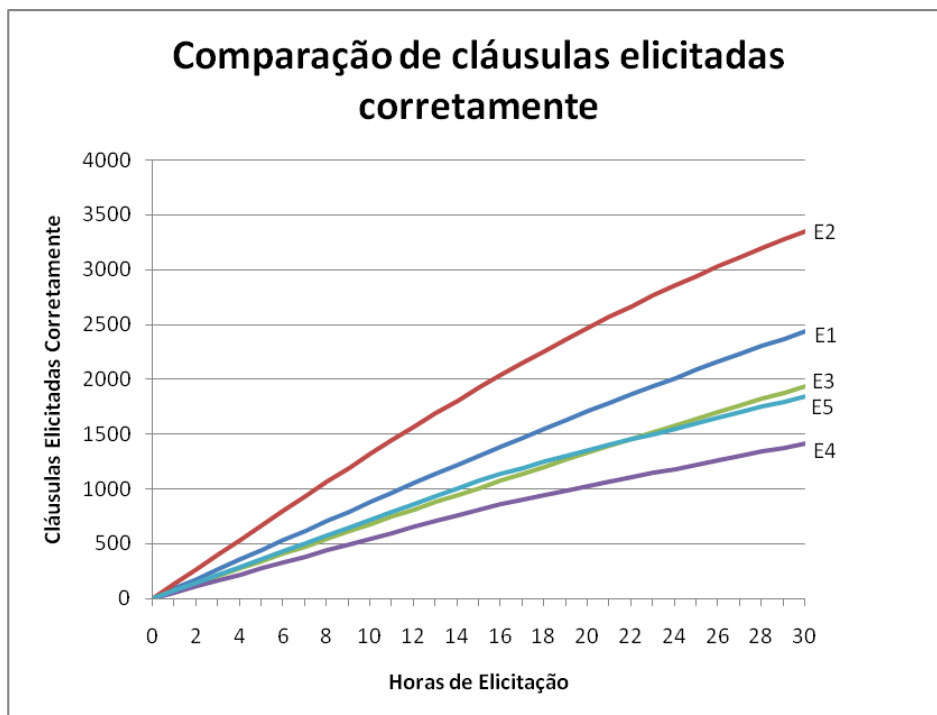
Fonte: Elaborada pelo autor.

As comparações entre os cenários estão representadas nas figuras a seguir. A Figura 4.11 mostra uma comparação quanto ao número total de cláusulas, divididas em cláusulas corretas, erradas e repetidas. A Figura 4.12 exibe uma comparação com base no número de cláusulas elicitadas corretamente. A Figura 4.13 apresenta uma comparação do custo total da elicitação. Por fim, a Figura 4.14 apresenta uma comparação de custo por cláusula correta, considerando o custo total da elicitação ao final das 30 horas dividido pelo número de cláusulas elicitadas corretamente no mesmo período.



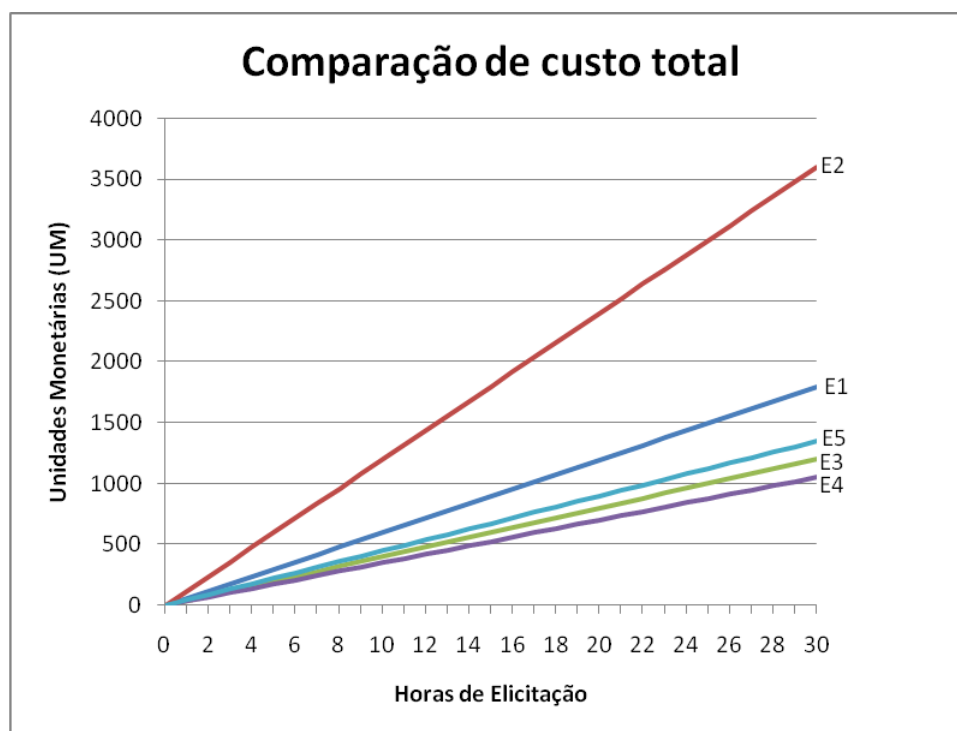
Fonte: Elaborada pelo autor

Figura 4.11. Comparação de cláusulas elicitadas



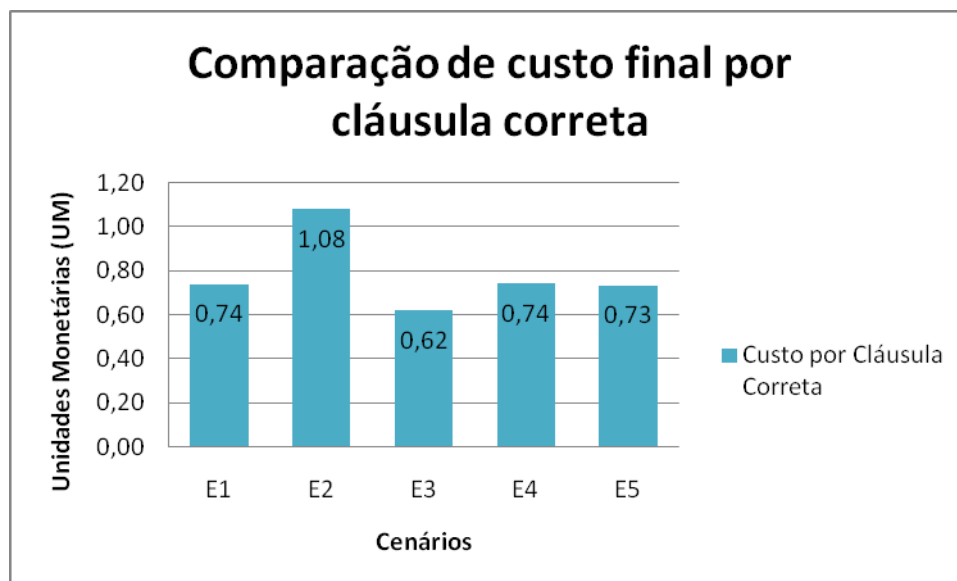
Fonte: Elaborada pelo autor

Figura 4.12. Comparação de cláusulas elicítadas corretamente



Fonte: Elaborada pelo autor

Figura 4.13. Comparação de custo total



Fonte: Elaborada pelo autor

Figura 4.14. Comparação de custo final por cláusula correta

Comparando os cenários E1 e E2, é possível observar que dobrando o número de profissionais na equipe (E2), o número total de cláusulas elicitadas não dobrou, mas aumentou em aproximadamente 80,2% (Figura 4.11), e o número de cláusulas corretas aumentou em 37,2% (Figura 4.12). A relação do número de profissionais na equipe com a produtividade não é linear devido ao aumento do número de cláusulas repetidas elicitadas e aumento no *overhead* de comunicação. O custo por cláusula correta em E2, ao final das 30 horas de elicitação, teve um aumento de 45,9% com relação a E1 (Figura 4.14).

Entre E2 e E3, observou-se que utilizando profissionais menos experientes (E3) foi possível reduzir o custo total da elicitação em 66,7% (Figura 4.13), porém reduziu-se em 42,1% o número de cláusulas elicitadas corretamente (Figura 4.12). Comparado à E1, apesar de E3 obter um maior número total de cláusulas, nota-se que o número de cláusulas elicitadas corretamente foi maior em E1 (Figura 4.11), mesmo utilizando metade do número de profissionais de E3. Essa superioridade de E1 é decorrente da maior produtividade dos profissionais mais experientes alocados nesse cenário.

A função do cenário E4 foi verificar se é válido substituir um *stakeholder* experiente por dois menos experientes, considerando que o custo dos menos experientes é menor. O custo total final de E4 foi 41,7% inferior à E1 (Figura 4.13), porém essa também foi a proporção em que se reduziu o número de cláusulas

elicitadas corretamente. Essa proporção pôde ser reforçada pela proximidade entre o custo por cláusula correta entre esses dois cenários, observada através da Figura 4.14.

O cenário E5 serviu como base de comparação com E4. Em E4 foram alocados um elicitador engenheiro e dois *stakeholders* pouco experientes. Em E5 inverteu-se as experiências dos profissionais, sendo considerados um elicitador estudante e dois *stakeholders* experientes. Em E5 obteve-se um aumento de 30,2% no número de cláusulas elicitadas corretamente em relação a E4 (Figura 4.12). O custo total teve um aumento de 28,6% (Figura 4.13).

A decisão de qual cenário aplicar depende de fatores que influenciam no projeto e na empresa em questão, principalmente da disponibilidade de profissionais, prazo do projeto e recursos disponíveis. O modelo em si não mostra qual a decisão gerencial que deve ser tomada, mas auxilia os gerentes a enxergarem o contexto dos cenários (BOEHM, 1981).

Considerando por exemplo o tempo para atingir as primeiras mil cláusulas a serem elicitadas no projeto, se uma empresa está com o prazo curto e dispõe de recursos poderia optar por E2, concluindo a elicitação das mil cláusulas em torno de 8 horas, mas assumindo um custo total de 960,00UM. Mas caso a empresa tenha tempo disponível e opte por economizar ao máximo, E3 seria o cenário escolhido, tendo um custo total de 600,00UM (aproximadamente 37,5% de redução em relação a E2), mas terminando a tarefa em 15 horas (87,5% a mais de tempo em relação a E2).

Caso a empresa tenha um prazo limitado para elicitação e desconheça o domínio do problema a ser abordado no projeto, deve-se evitar alocar profissionais com pouca experiência, principalmente *stakeholders*, pois, apesar da redução considerável no custo do projeto, há um risco maior, visto que algumas cláusulas deixadas de fora pelos profissionais podem representar um conhecimento essencial para o projeto.

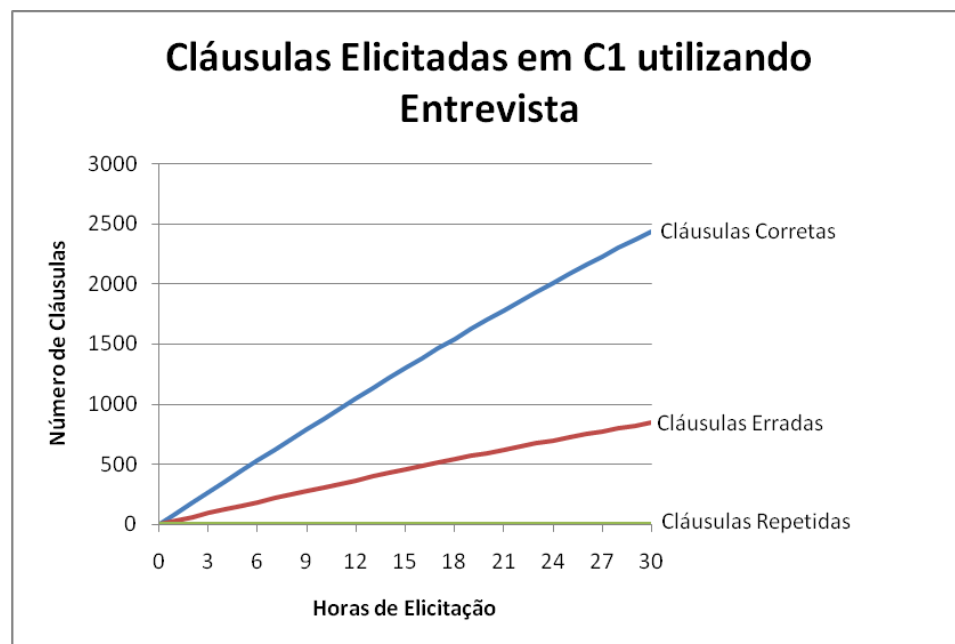
4.2 Comparação entre as técnicas de elicitação

Nesta seção serão considerados dois cenários. Um cenário mais simples e outro mais complexo. Cada cenário será simulado utilizando cada uma das quatro técnicas de elicitação. Ao final de cada cenário será realizada uma comparação entre as técnicas.

4.2.1 Cenário C1

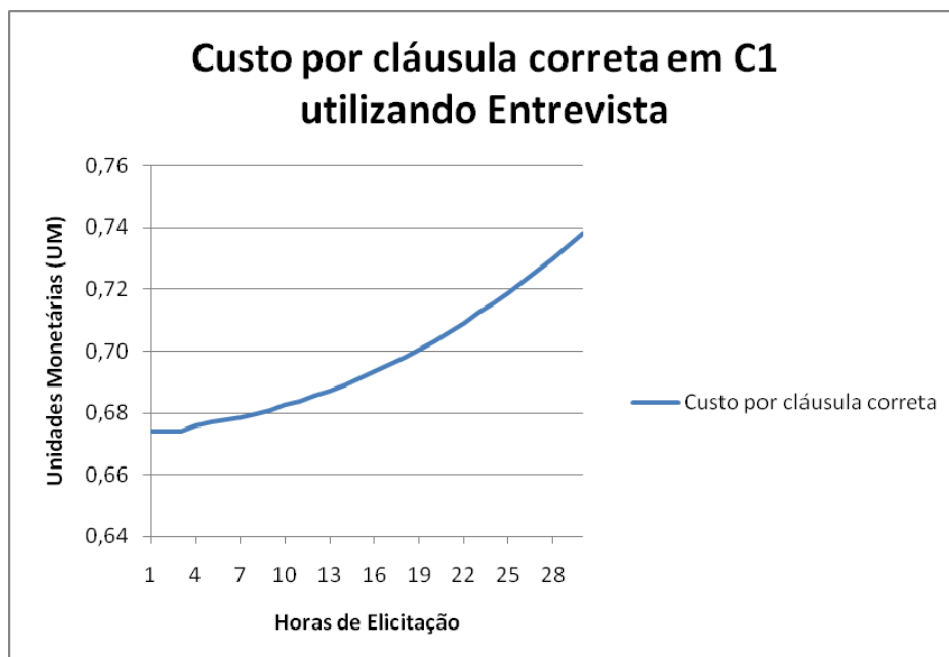
C1 é o mesmo cenário apresentado em E1 (Seção 4.1.1). Em C1 foram alocados 1 elicitor engenheiro de software/analista e 1 *stakeholder* experiente.

As figuras abaixo mostram os resultados da simulação, onde para cada técnica existem dois gráficos: um exibindo as cláusulas elicidadas e outro apresentando o custo por cláusula correta ao longo da elicitação.



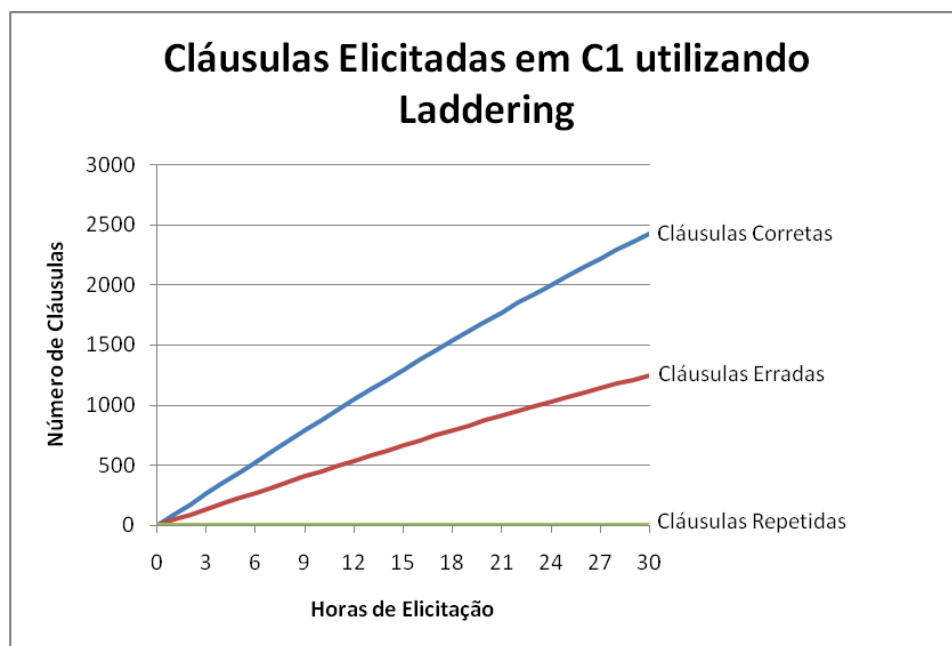
Fonte: Elaborada pelo autor

Figura 4.15. Cláusulas elicidadas em C1 utilizando a técnica de Entrevista



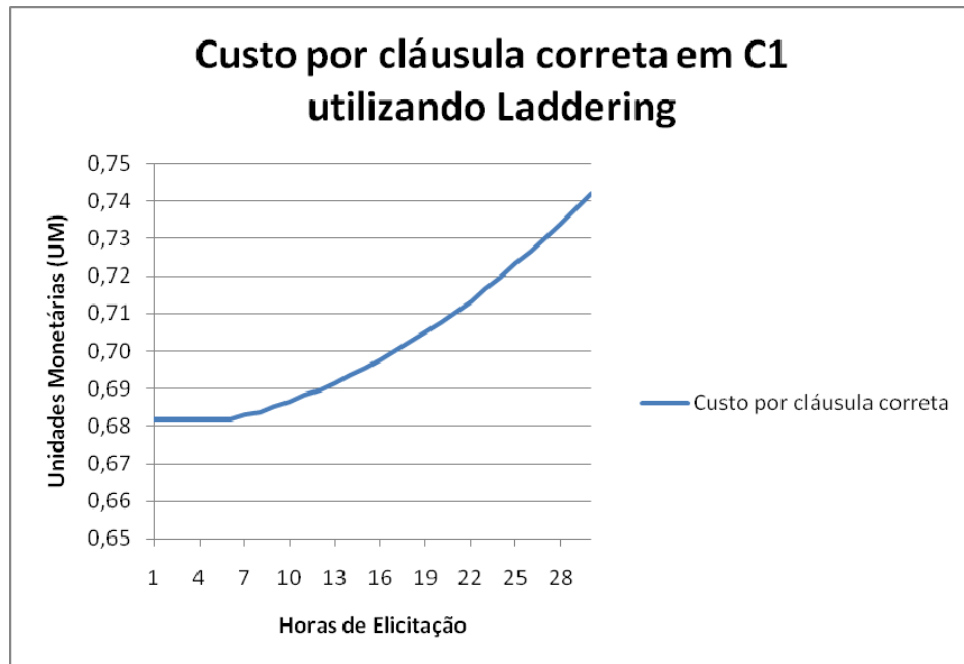
Fonte: Elaborada pelo autor

Figura 4.16. Custo por cláusula correta em C1 utilizando a técnica de Entrevista



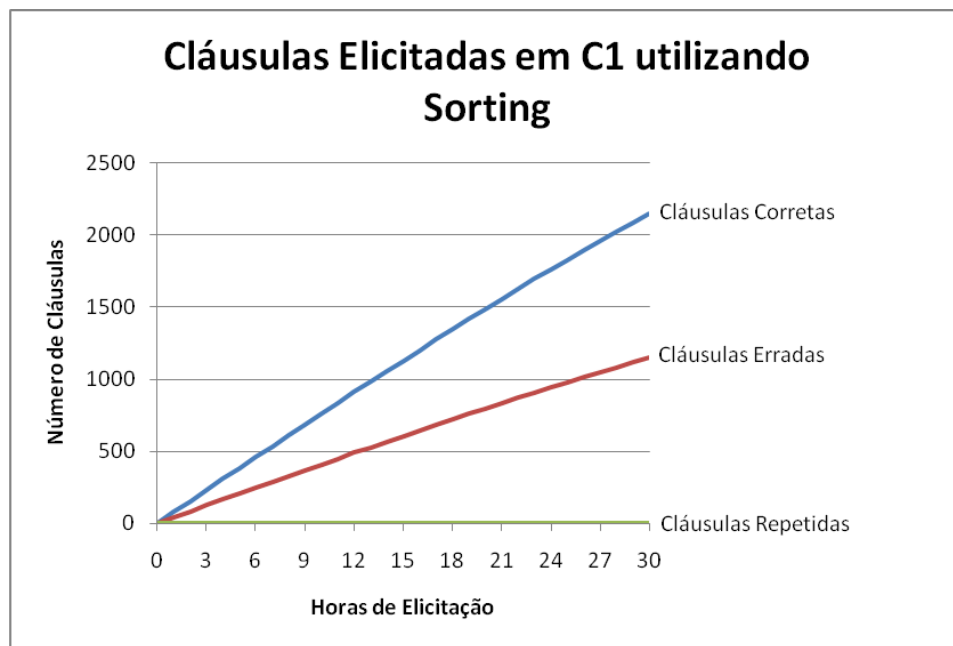
Fonte: Elaborada pelo autor

Figura 4.17. Cláusulas elicítadas em C1 utilizando a técnica de *Laddering*



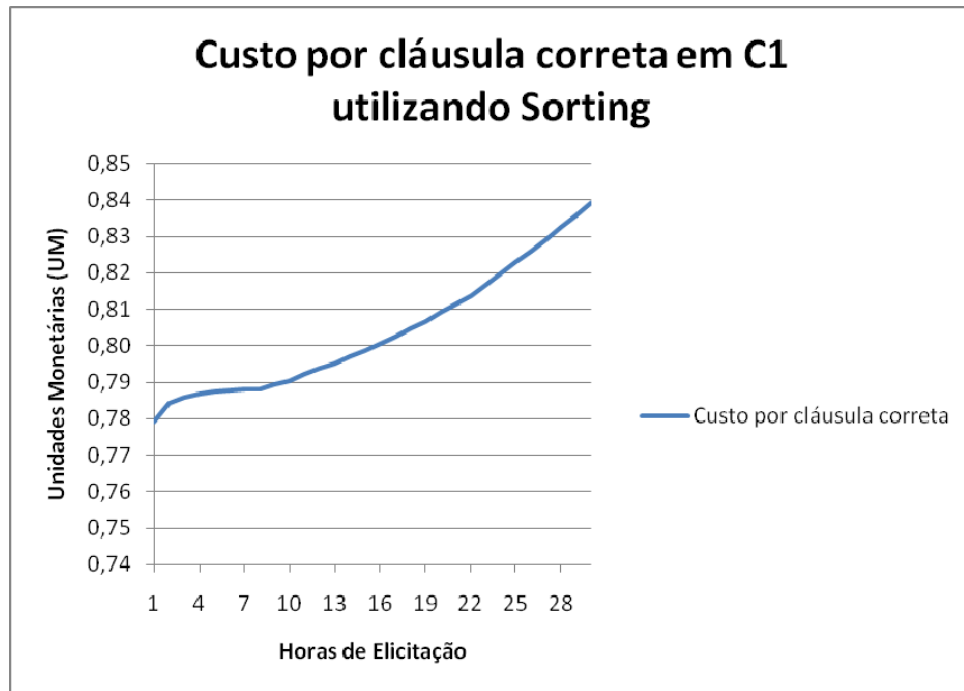
Fonte: Elaborada pelo autor

Figura 4.18. Custo por cláusula correta em C1 utilizando a técnica de *Laddering*



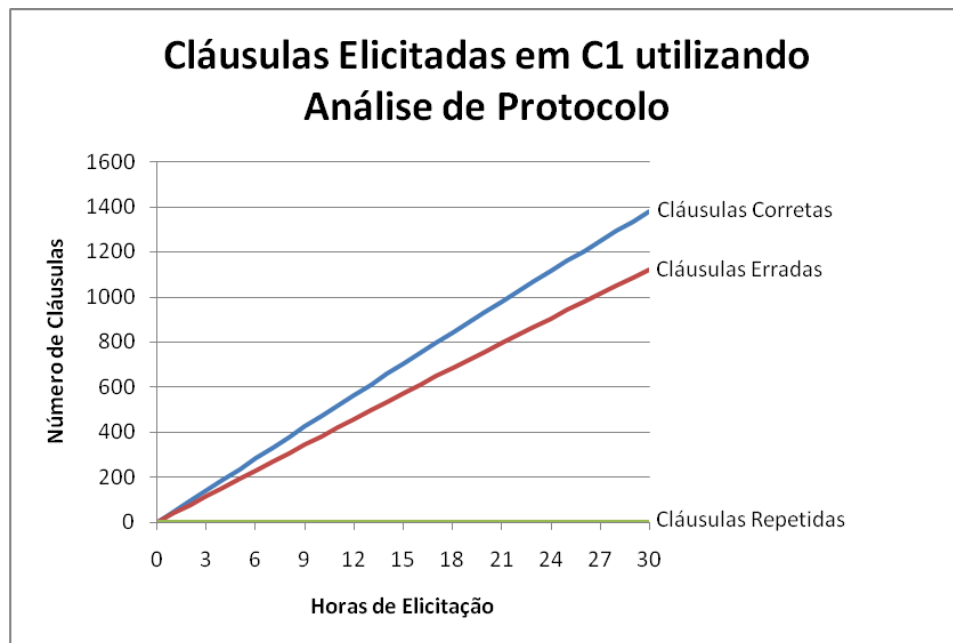
Fonte: Elaborada pelo autor

Figura 4.19. Cláusulas elicítadas em C1 utilizando a técnica de *Sorting*



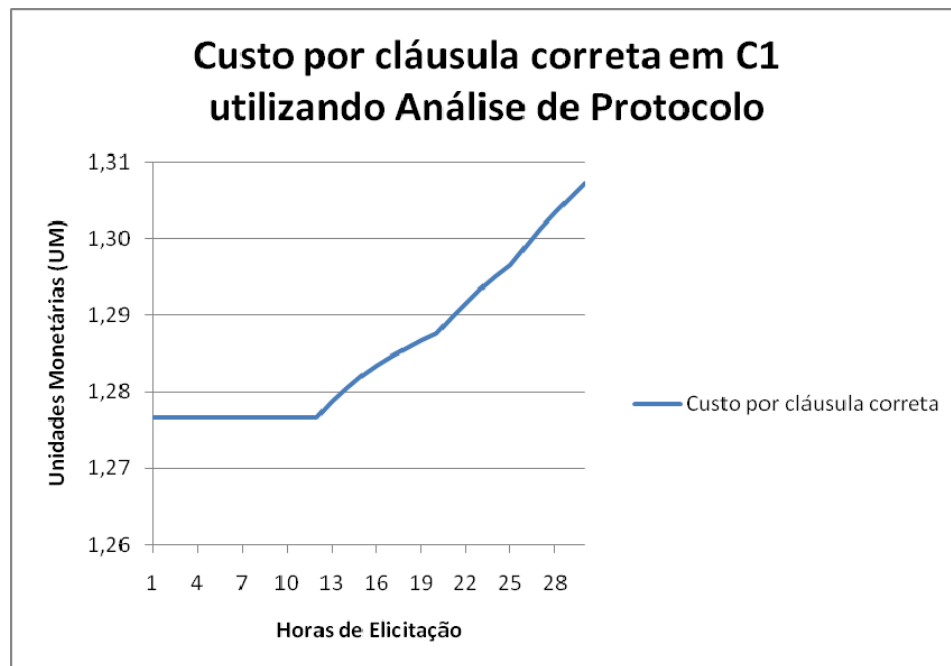
Fonte: Elaborada pelo autor

Figura 4.20. Custo por cláusula correta em C1 utilizando a técnica de *Sorting*



Fonte: Elaborada pelo autor

Figura 4.21. Cláusulas elicítadas em C1 utilizando a técnica de Análise de Protocolo

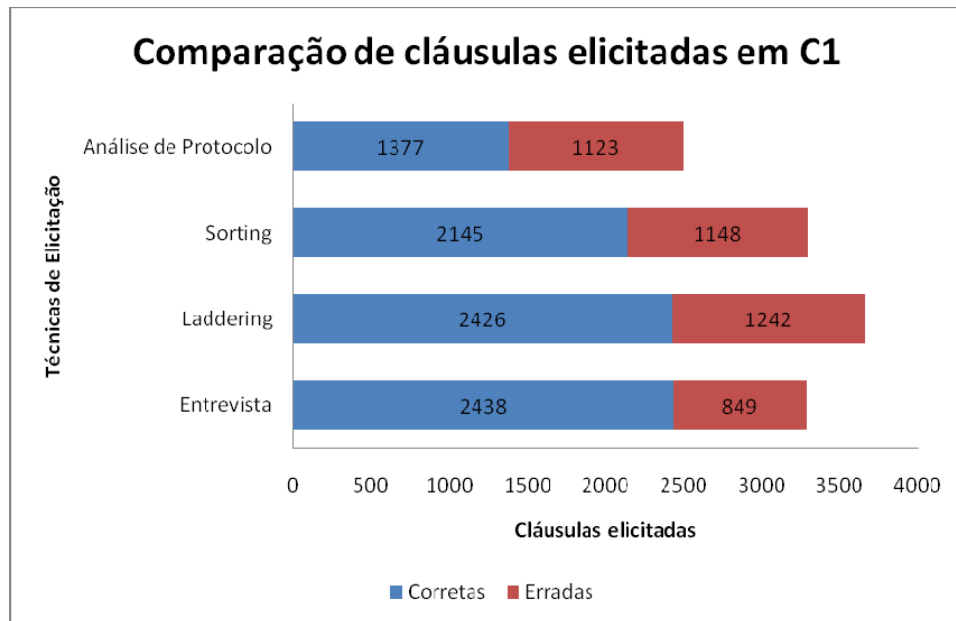


Fonte: Elaborada pelo autor

Figura 4.22. Custo por cláusula correta em C1 utilizando a técnica de Análise de Protocolo

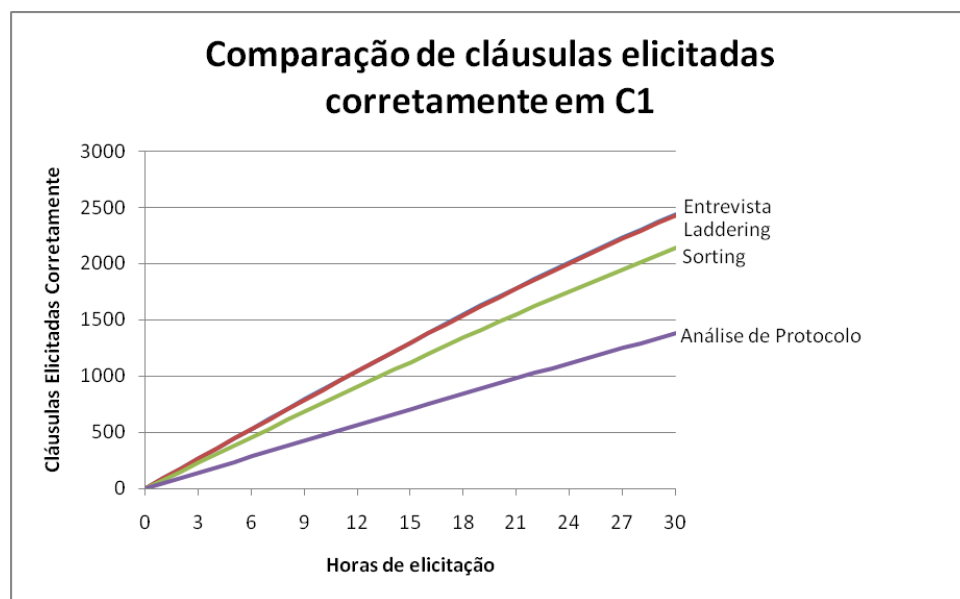
Uma observação importante, é que no cenário C1, a variável que influencia na curva do custo por cláusula correta, não permitindo que ela seja uma reta horizontal (custo fixo), é a "dificuldade em encontrar novos requisitos". No caso de C1, essa variável começa a impactar de maneira mais perceptível a partir de 600 cláusulas elicítadas corretamente, quando essa dificuldade passa a afetar a produtividade. Nos gráficos referentes à utilização das técnicas mais efetivas, é possível observar uma mudança na curva mais cedo. Já na técnica de análise de protocolo, essa mudança na curva só ocorre pouco depois de 12 horas na simulação, quando finalmente a marca das 600 cláusulas elicítadas corretamente é atingida.

As três figuras a seguir apresentam uma comparação entre as técnicas de elicitação com relação ao cenário C1. A Figura 4.23 compara as quatro técnicas com relação ao número total de cláusulas elicítadas. A Figura 4.24 compara as técnicas no que diz respeito ao total de cláusulas elicítadas corretamente durante a elicitação. Por fim, a Figura 4.25 mostra a comparação do custo por cláusula correta ao final das trinta horas de sessões de elicitação.



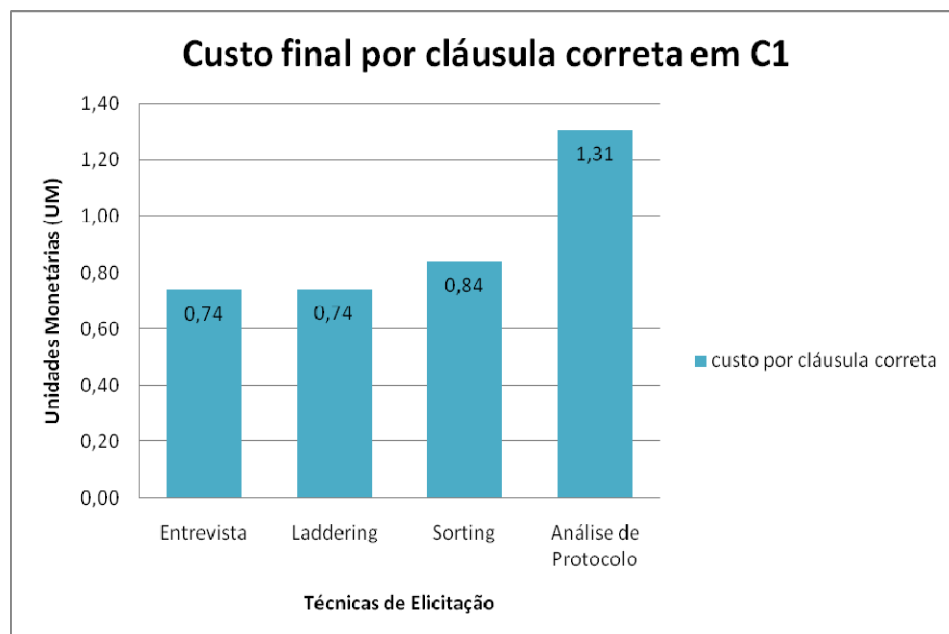
Fonte: Elaborada pelo autor

Figura 4.23. Comparação de cláusulas elicítadas em C1



Fonte: Elaborada pelo autor

Figura 4.24. Comparação de cláusulas elicítadas corretamente em C1



Fonte: Elaborada pelo autor

Figura 4.25. Comparação de custo final por cláusula correta em C1

Através da Figura 4.23 é possível observar que o uso da técnica de *laddering* permitiu elicitar um maior número de cláusulas durante as 30 horas de elicitação (11,6% a mais do que a entrevista, 11,4% a mais que *sorting* e 46,7% a mais que análise de protocolo).

Considerando apenas as cláusulas corretas (Figura 4.23 e Figura 4.24), a técnica de entrevista foi melhor (0,5% melhor que *laddering*, 13,7% melhor que *sorting* e 77,1% melhor que análise de protocolo). Devido à proximidade das curvas de entrevista e *laddering* na Figura 4.24, a curva referente a entrevista ficou pouco visível no gráfico.

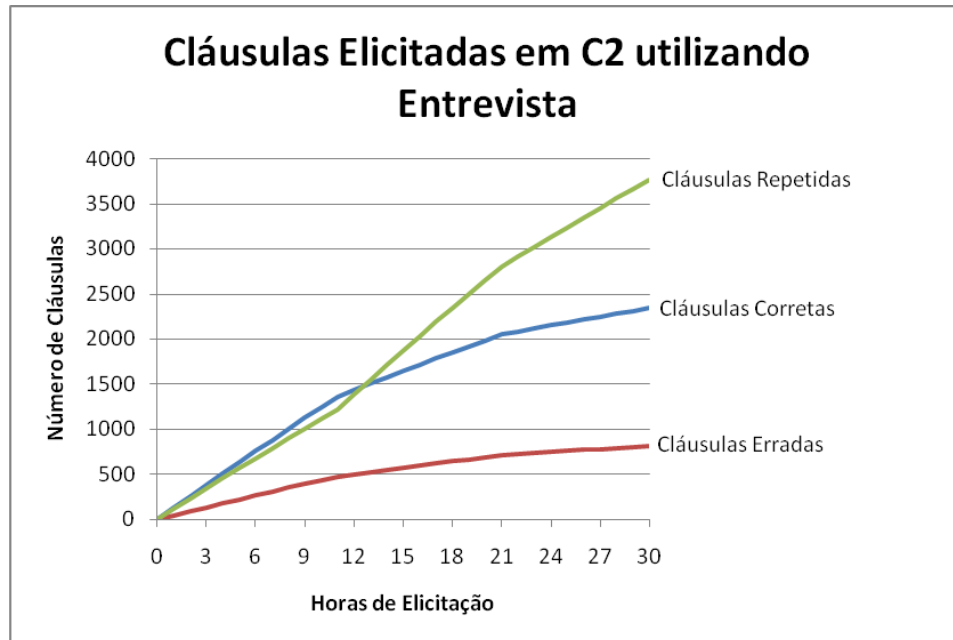
Entrevista e *laddering* apresentaram o menor custo por cláusula correta ao final das 30 horas de sessões de elicitação (Figura 4.25). Esse custo foi 11,9% menor que o custo por cláusula de *sorting* e 43,5% menor que análise de protocolo.

Analisando-se os gráficos de cláusulas elicítadas em cada técnica, é possível observar que na análise de protocolo (Figura 4.21) as curvas de cláusulas corretas e erradas ficaram mais próximas do que nas demais técnicas. Ao final das 30 horas de elicitação, as cláusulas erradas corresponderam a 45% do total de cláusulas elicítadas. A técnica que foi melhor nesse quesito foi a entrevista, onde as cláusulas erradas representaram 26% do total de cláusulas elicítadas.

4.2.2 Cenário C2

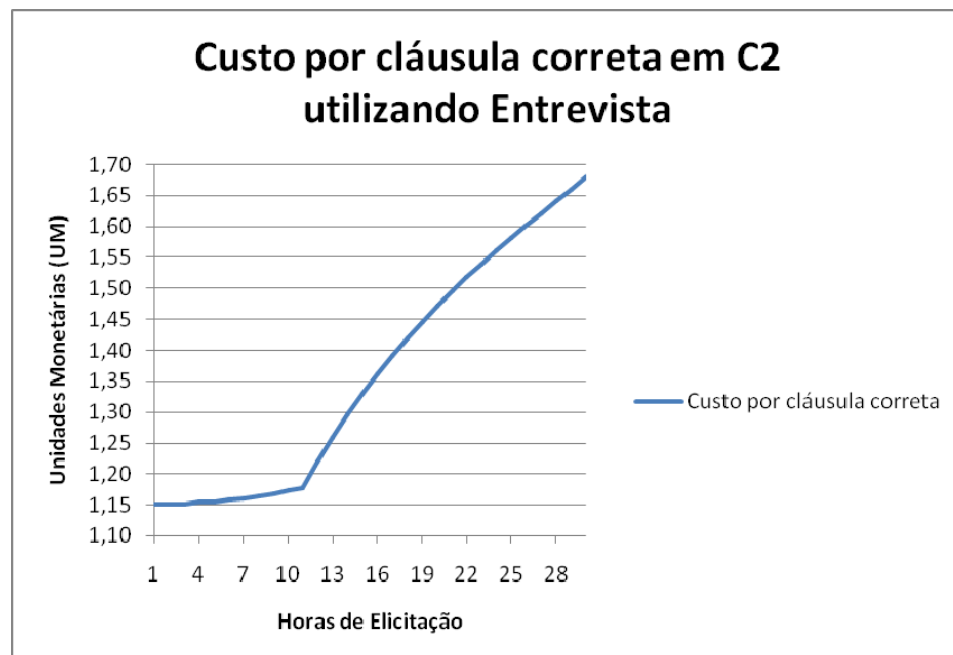
No cenário C2 foram alocados 2 elicitadores engenheiro de software/analista, 1 elicitador estudante, 6 *stakeholders* experientes e 2 *stakeholders* pouco experientes.

As figuras a seguir mostram os resultados das simulações para cada técnica.



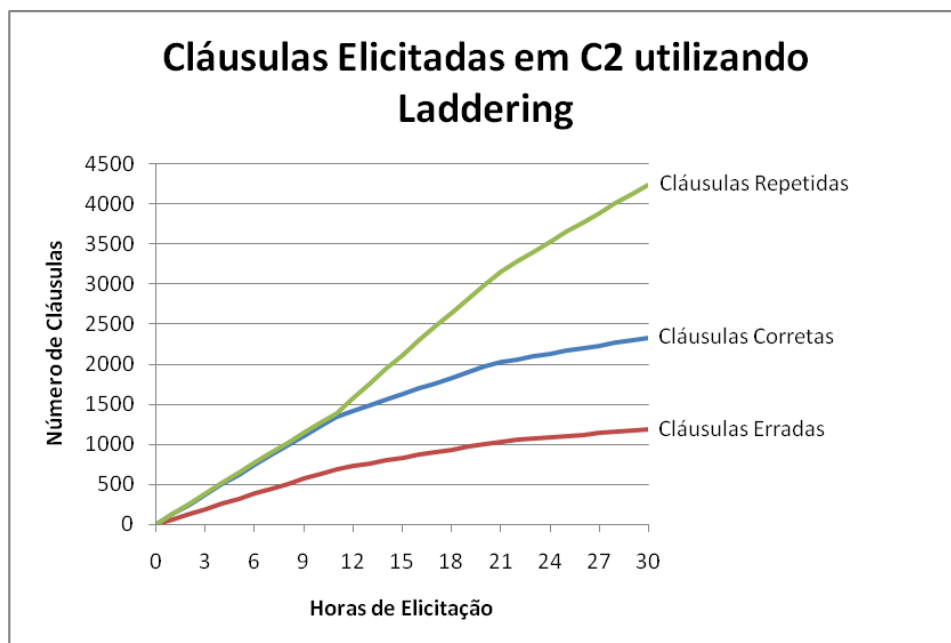
Fonte: Elaborada pelo autor

Figura 4.26. Cláusulas elicítadas em C2 utilizando a técnica de Entrevista



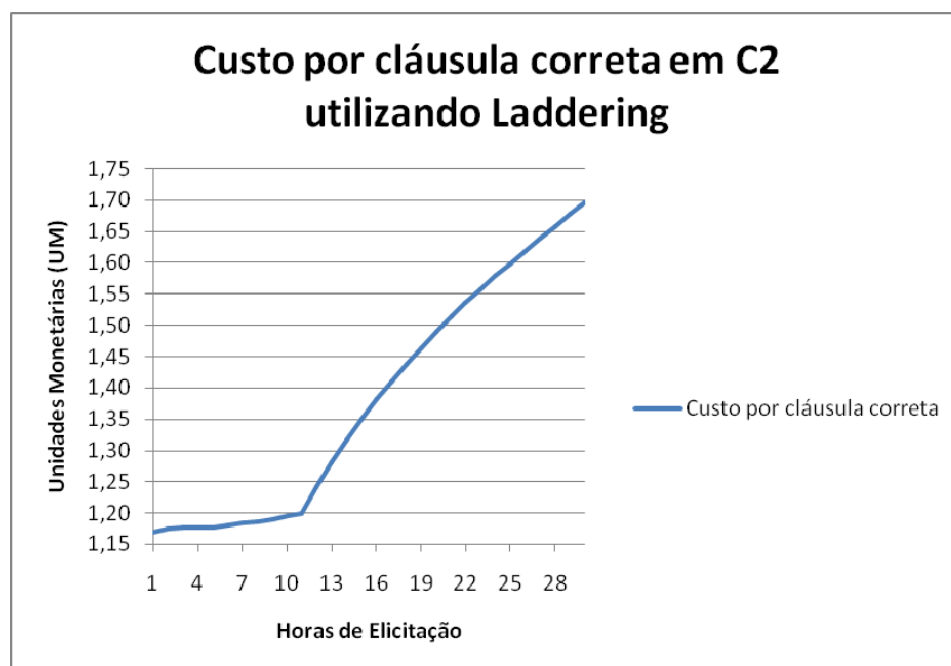
Fonte: Elaborada pelo autor

Figura 4.27. Custo por cláusula correta em C2 utilizando a técnica de Entrevista



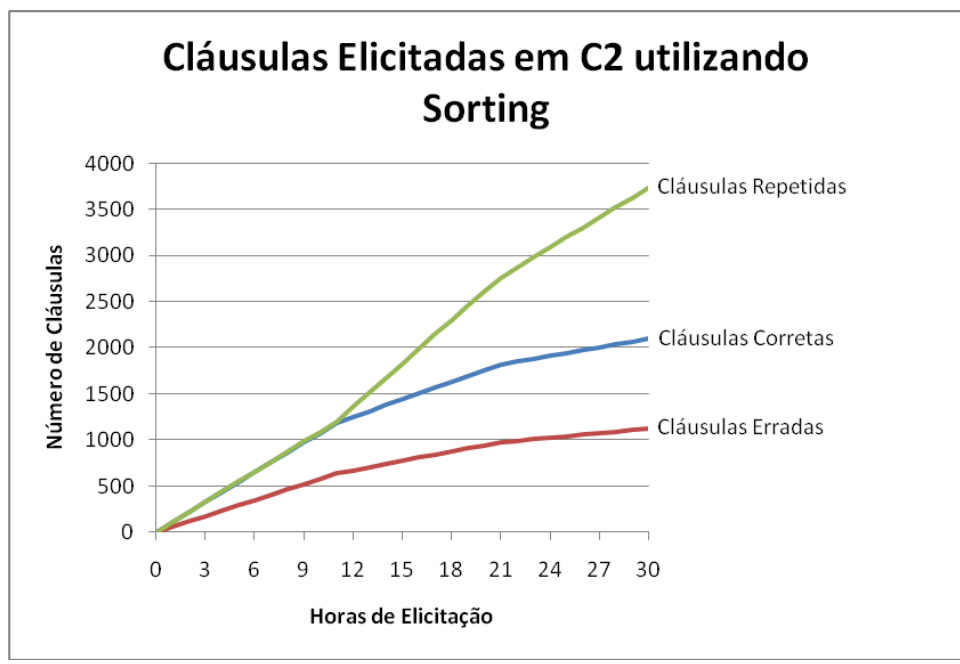
Fonte: Elaborada pelo autor

Figura 4.28. Cláusulas elicítadas em C2 utilizando a técnica de *Laddering*



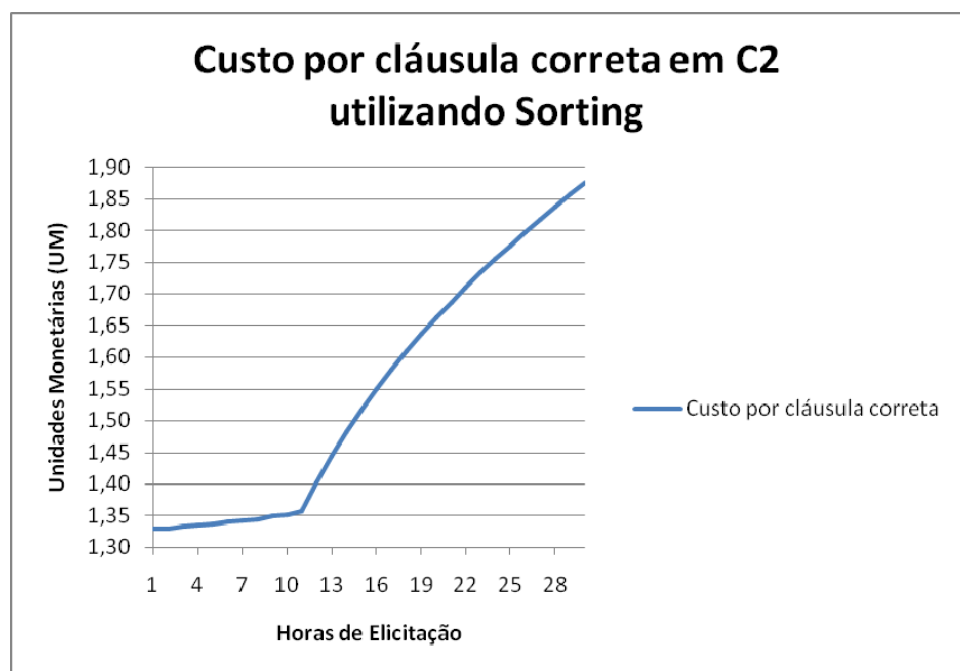
Fonte: Elaborada pelo autor

Figura 4.29. Custo por cláusula correta em C2 utilizando a técnica de *Laddering*



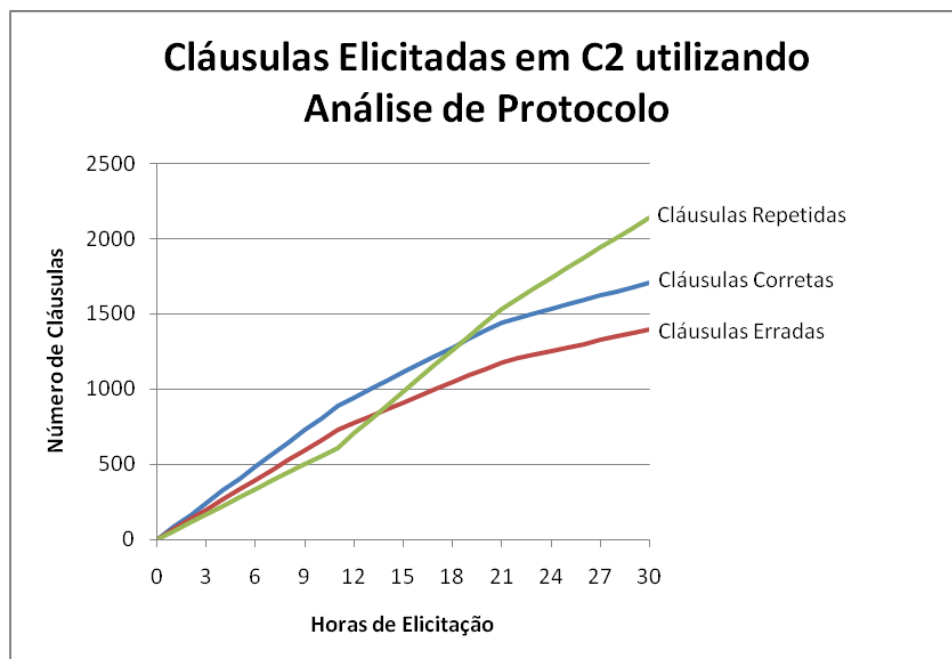
Fonte: Elaborada pelo autor

Figura 4.30. Cláusulas elicítadas em C2 utilizando a técnica de *Sorting*



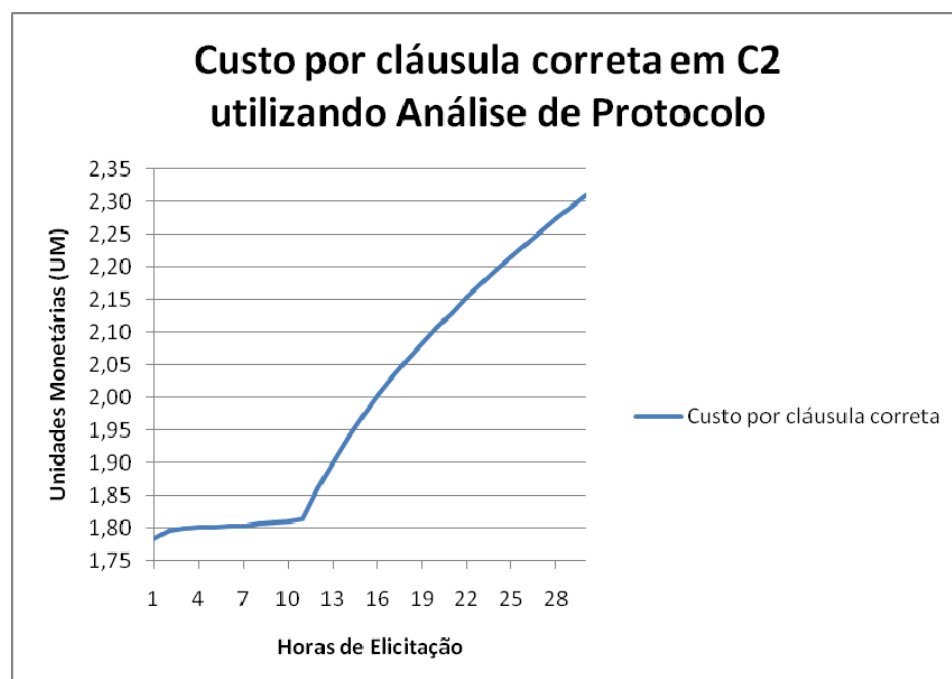
Fonte: Elaborada pelo autor

Figura 4.31. Custo por cláusula correta em C2 utilizando a técnica de *Sorting*



Fonte: Elaborada pelo autor

Figura 4.32. Cláusulas elicítadas em C2 utilizando a técnica de Análise de Protocolo



Fonte: Elaborada pelo autor

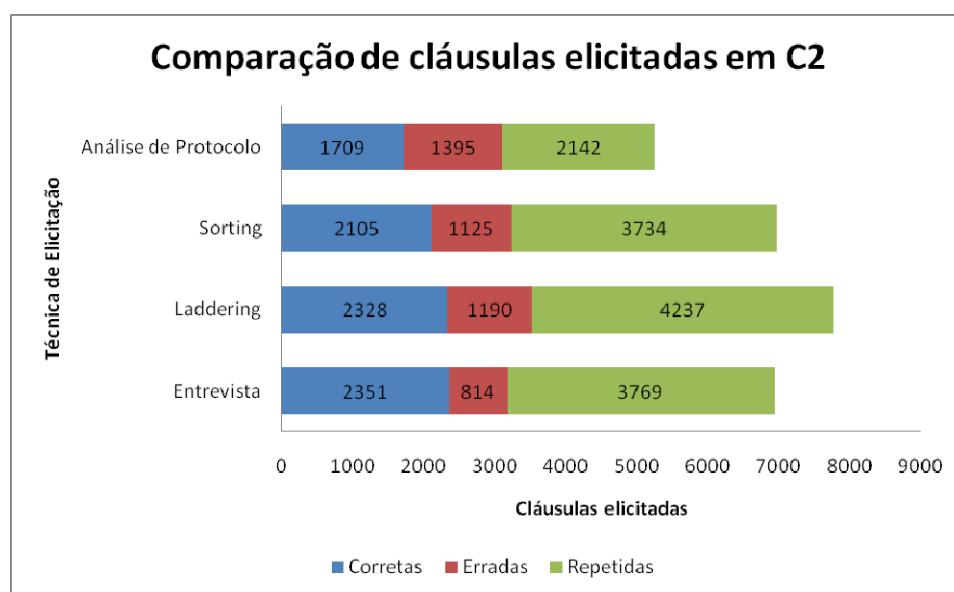
Figura 4.33. Custo por cláusula correta em C2 utilizando a técnica de Análise de Protocolo

No caso do cenário C2, a variável que mais influencia no custo por requisito correto (impedindo que o custo seja fixo) é o número de cláusulas repetidas. Esse número aumenta muito a partir do momento que os três elicítadores terminam de elicitar cláusulas dos três primeiros *stakeholders* e começam a elicitar de novos três.

Na simulação, isso ocorre por volta de 10 horas de elicitação. É perceptível nos gráficos de todas as técnicas que o número de cláusulas repetidas aumenta muito a partir dessa hora e, conseqüentemente, assim também acontece com o custo por requisito correto.

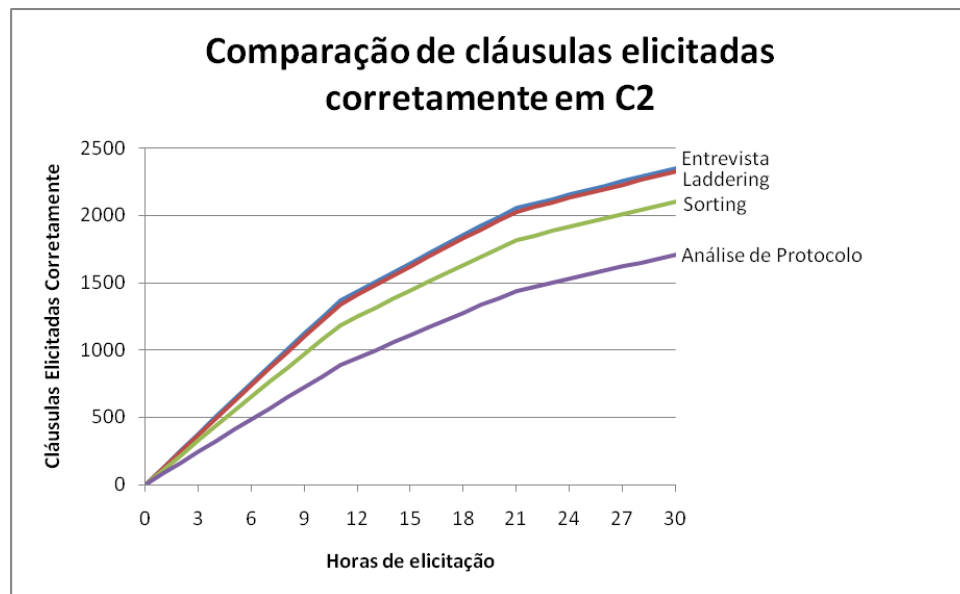
As curvas referentes às simulações de C2 também têm influência da variável "dificuldade de encontrar novos requisitos". Porém, no caso desse cenário, essa influência é muito menos perceptível do que a influência dos requisitos repetidos, que foi explicada no parágrafo anterior.

As figuras a seguir apresentam uma comparação entre as quatro técnicas de elicitação com relação ao cenário C2. A Figura 4.34 compara as técnicas com relação ao número total de cláusulas elicítadas. A Figura 4.35 compara as técnicas quanto ao total de cláusulas elicítadas corretamente durante a elicitação. Por fim, a Figura 4.36 mostra a comparação do custo por cláusula correta ao final das trinta horas de sessões de elicitação.



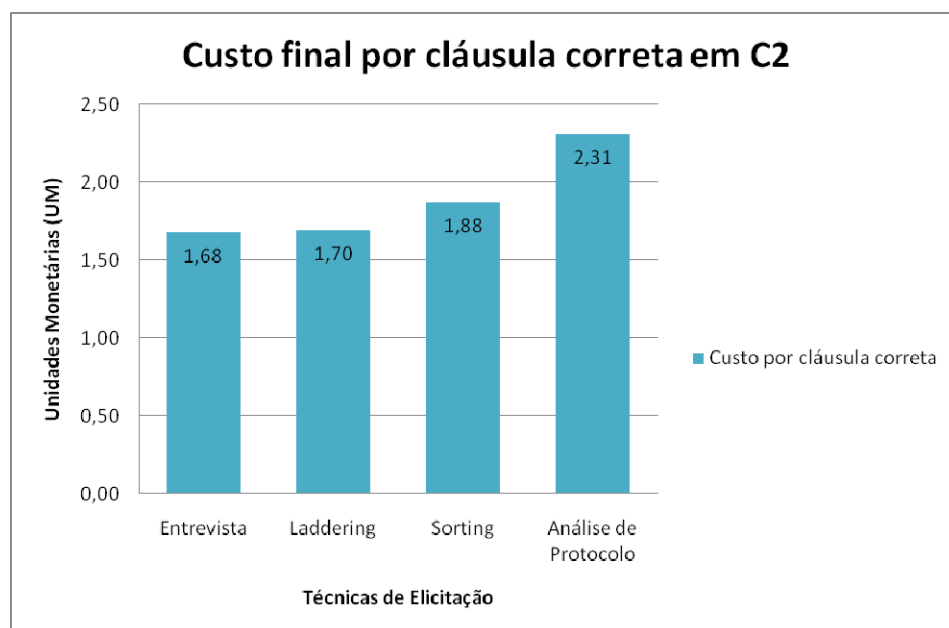
Fonte: Elaborada pelo autor

Figura 4.34. Comparação de cláusulas elicítadas em C2



Fonte: Elaborada pelo autor

Figura 4.35. Comparação de cláusulas elicítadas corretamente em C2



Fonte: Elaborada pelo autor

Figura 4.36. Comparação de custo final por cláusula correta em C2

No cenário C2, as cláusulas repetidas tiveram um grande destaque, sendo responsáveis pela maior parte das cláusulas elicítadas ao final das 30 horas de elicitação. Isso se deve ao elevado número de *stakeholders* alocados. Observando os gráficos referentes a cada técnica, é possível verificar que o número de cláusulas repetidas vai aumentando de maneira mais branda do início da elicitação até o fim da sessão com os primeiros *stakeholders* alocados (em torno de 10 horas). Quando

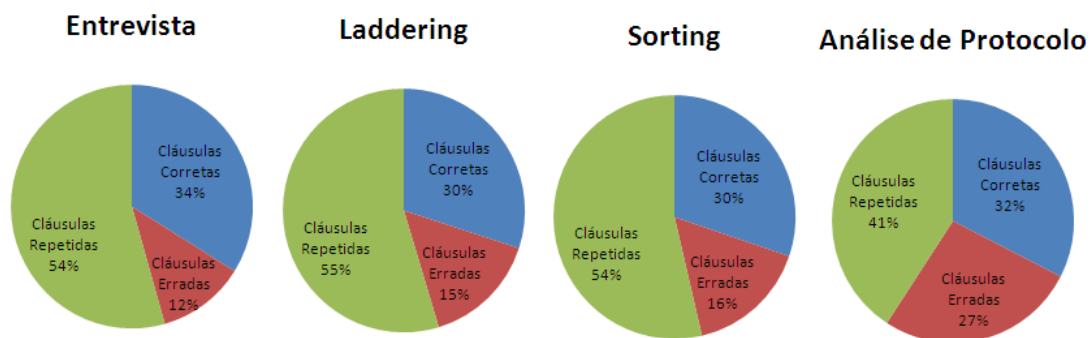
novos elicitadores são alocados, eles não sabem ao certo o que já foi elicitado a partir dos *stakeholders* anteriormente alocados. Isso faz com que a taxa de requisitos repetidos aumente de maneira muito mais brusca. Dessa forma, a taxa de elicitação de requisitos corretos cai e, conseqüentemente, aumenta o custo por requisito correto, o que também é visível através das figuras de custo por requisito correto referentes a cada técnica.

Assim como no cenário C1, em C2 através do uso da técnica de *laddering* foi possível elicitar um maior número total de cláusulas (Figura 4.34). Em seguida vieram as técnicas de *sorting* (10,2% a menos), entrevista (10,6% a menos) e análise de protocolo (32,4% a menos).

Levando-se em consideração apenas as cláusulas corretas (Figura 4.34 e Figura 4.35), a técnica de entrevista foi a melhor, seguida de *laddering* (1% a menos de cláusulas corretas), *sorting* (10,5% a menos) e análise de protocolo (27,3% a menos).

A técnica de entrevista apresentou o menor custo por requisito correto ao final das 30 horas de elicitação (Figura 4.36). Em seguida vieram as técnicas de *laddering*, tendo o custo 1,2% maior do que entrevista; *sorting*, 11,9% maior; e, por último, análise de protocolo, apresentando um custo por requisito correto 37,5% maior que entrevista.

Considerando o esforço efetivo realizado em cada técnica, ou seja, a porcentagem do esforço gasto durante a elicitação que foi utilizado para elicitar cláusulas corretas, a técnica de entrevista foi a melhor, tendo 34% dos requisitos elicitados corretamente ao final das 30 horas de sessões de elicitação. Em segundo lugar apareceu a técnica de análise de protocolo, com 32% das cláusulas corretas. As técnicas de *laddering* e *sorting* apresentaram 30% das cláusulas corretas ao final das sessões de elicitação. A Figura 4.37 mostra maiores detalhes sobre as relações das porcentagens de cláusulas elicitadas em cada técnica ao final das 30 horas de elicitação.



Fonte: Elaborada pelo autor

Figura 4.37. Comparação das porcentagens de cláusulas elicitadas em C2

No geral (tanto no cenário C1 quanto em C2), considerando as cláusulas corretas, os resultados das simulações foram condizentes com os resultados dos estudos (DIESTE; JURISTO, 2010; DIESTE; LOPEZ; RAMOS, 2008; DAVIS *et al.*, 2006). A técnica de entrevista foi a mais efetiva, seguida da técnica de *laddering*, que apresentou praticamente a mesma efetividade da entrevista. A técnica de *sorting* foi a terceira mais efetiva; e a técnica de análise de protocolo apresentou os piores resultados, sendo muito menos efetiva que as demais.

5 PORTAL CORPORATIVO PARA APOIO A DECISÃO

Com o intuito de facilitar o acesso das empresas de software aos modelos produzidos neste trabalho, foi desenvolvido um portal corporativo para apoio a decisão. A arquitetura do portal foi proposta durante o presente trabalho e o desenvolvimento foi realizado por estudantes⁹ da graduação, via projetos de iniciação científica com bolsas fornecidas pela Fapemig.

O portal é uma ferramenta de apoio aos gerentes de projetos, disponível através da Internet, na qual é possível cadastrar as empresas gratuitamente, configurar cenários de acordo com o perfil da empresa e dos projetos, e executar as simulações remotamente. A seguir serão apresentadas as principais telas do portal.

A Figura 5.1 mostra a tela inicial do portal, onde os gerentes ou responsáveis pela empresa entram no sistema com seu usuário e senha. Nessa tela também é possível acessar a opção para cadastrar uma nova empresa no portal, através do link "Cadastre sua empresa".



Fonte: Elaborada pelo autor

Figura 5.1. Tela inicial do portal corporativo

⁹ Os estudantes responsáveis pelo desenvolvimento do Portal foram Jean P. Freitas e Jailton J. Sousa Coelho

No sistema, existem dois níveis de usuário. O primeiro é o responsável pela empresa, que faz o cadastro no portal e pode criar, editar ou excluir projetos da empresa. O segundo nível de usuário são os gerentes de projetos, que possuem usuários e senhas relacionados a um ou mais projetos específicos. Os usuários gerentes podem editar os projetos pelos quais são responsáveis, configurando o cenário do projeto e executando simulações.

Na Figura 5.1, ao clicar em “Cadastre sua empresa”, o responsável pela empresa no portal deverá entrar com dados da empresa e escolher usuário e senha para realizar o acesso ao portal. A Figura 5.2 mostra a tela de cadastro de empresas.

PORTAL CORPORATIVO

Nome da Empresa:

CNPJ:

E-mail:

Telefone:

Responsável pelo Cadastro:

Escolha um Login:

Escolha uma Senha:

Digite novamente a Senha:

OBS.:Esse Login e Senha são apenas para uso do responsável pelo Cadastro da Empresa. Novos Logins e Senhas serão distribuídos para cada projeto cadastrado pela empresa.

Fonte: Elaborada pelo autor

Figura 5.2. Tela de cadastro de empresas

Na tela inicial (Figura 5.1), uma vez que o responsável pela empresa entre no sistema com seu usuário e senha, é exibida a tela de projetos (Figura 5.3). Nessa tela o usuário responsável pela empresa pode cadastrar novos projetos e abrir projetos já existentes para executar simulações.



Fonte: Elaborada pelo autor

Figura 5.3. Tela de projetos

Na tela de projetos, exibida na Figura 5.3, clicando-se em “Cadastre um novo Projeto da sua empresa”, é exibida a tela de cadastro de projetos (Figura 5.4). No cadastro de um projeto é definido um novo usuário e senha próprios do projeto. Dessa forma, cada gerente da empresa pode ter um usuário cadastrado responsável apenas pelos projetos no qual trabalha.

The image shows a web application interface for 'PORTAL CORPORATIVO'. The form is titled 'PORTAL CORPORATIVO' at the top. It contains several input fields: 'Nome do Projeto:', 'Responsável pelo Projeto:', 'Escolha um Login:', 'Escolha uma Senha:', and 'Digite novamente a Senha:'. Each field is followed by a white input box. At the bottom of the form is a button labeled 'Cadastrar Projeto'.

Fonte: Elaborada pelo autor

Figura 5.4. Tela de cadastro de projetos

Quando um gerente de projetos entra na tela mostrada na Figura 5.1 com usuário e senha de gerente, ele é direcionado diretamente para a tela do projeto associado ao seu usuário. Essa tela é exibida na Figura 5.5 e é acessível também

pelos responsáveis pela empresa, através da tela de projetos (Figura 5.3), ao selecionar um projeto e clicar em “Abrir”.



Fonte: Elaborada pelo autor

Figura 5.5. Tela principal de um projeto

Na tela principal do projeto (Figura 5.5), é possível acessar o painel de controle para configurar as variáveis do cenário, realizar simulações a partir dos dados salvos no painel de controle e visualizar os gráficos de resultados.

O painel de controle é apresentado na Figura 5.6. Nessa tela, o usuário pode configurar as variáveis através dos botões “+” e “-”, ou ainda entrando com o valor diretamente nos campos correspondentes. Para saber o significado de uma variável, deve-se clicar sobre o símbolo da interrogação (?). Será então exibido um breve comentário sobre a variável em questão. Ao finalizar a configuração de todas as variáveis, o usuário deve clicar em “Salvar” e será então retornado para a tela anterior (Figura 5.5), onde poderá clicar em “Simular e visualizar os resultados”.

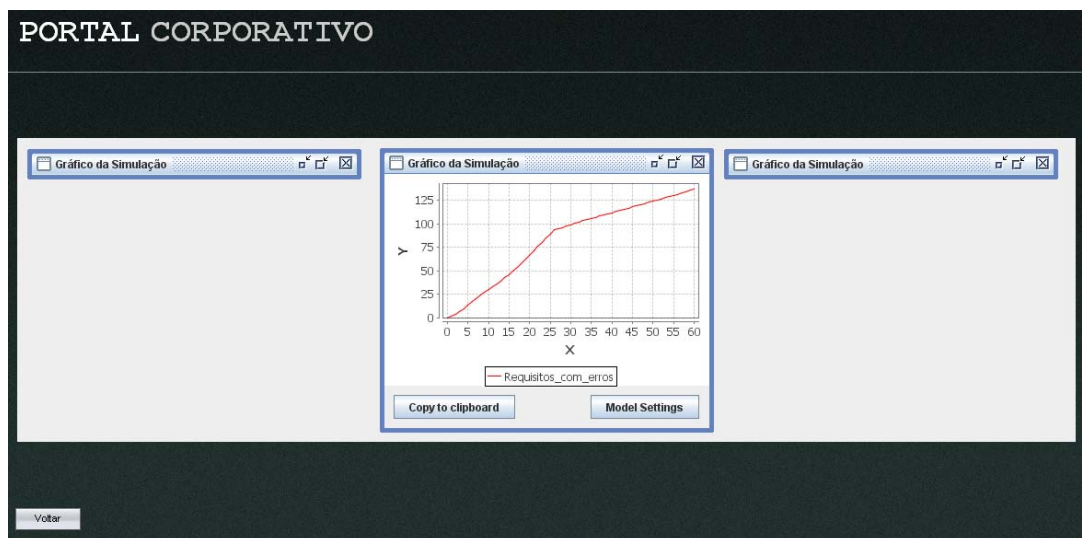


Fonte: Elaborada pelo autor

Figura 5.6. Tela do painel de controle

Após configurar as variáveis no painel de controle e clicar na opção “Simular e visualizar os resultados”, uma simulação baseada nas variáveis configuradas pelo usuário será executada. Ao final da simulação, os gráficos de resultados serão exibidos, permitindo que os gerentes analisem os resultados das suas configurações. A Figura 5.7 exibe um exemplo de resultado obtido a partir de uma simulação. Os

gráficos podem ser minimizados e maximizados para melhor visualização (na figura são mostrados dois gráficos minimizados). Além disso, pode ser feito um *zoom* no gráfico para visualizar maiores detalhes e o gráfico pode ser copiado através do botão “*Copy to clipboard*”. Algumas configurações de visualização estão disponíveis através do botão “*Model Settings*”.



Fonte: Elaborada pelo autor

Figura 5.7. Tela de resultados de simulações

O portal estará disponível em breve para uso público no endereço www.les.dpi.ufv.br.

6 CONCLUSÕES E TRABALHOS FUTUROS

A partir da pesquisa realizada, foi proposto um modelo de dinâmica de sistemas para a atividade de elicitação do processo de engenharia de requisitos de software. Esse modelo permite a configuração e simulação de cenários por gerentes de projetos para auxiliá-los na análise e tomadas de decisão relativas aos projetos de software. Dessa forma, o objetivo geral proposto nesse trabalho foi atingido.

Para desenvolver o modelo de dinâmica de sistemas, foi necessário isolar e analisar diversas variáveis que influenciam na elicitação, assim como os principais relacionamentos entre elas. Para auxiliar na seleção das variáveis, foi utilizado o framework GQM. Para estudar as principais relações entre elas, foram utilizados diagramas de influência. Esses diagramas foram então transformados em um diagrama estoque-fluxo, seguindo um processo disponível em (MADACHY, 2007).

O modelo de dinâmica de sistemas foi construído embasado em dados disponíveis na literatura, garantindo sua consistência. Dentre vários trabalhos investigados, não foi encontrado um modelo de dinâmica de sistemas que abordasse a atividade de elicitação de requisitos.

Os cenários apresentados no Capítulo 4 mostraram como é possível extrair informações relevantes a partir de simulações do modelo produzido neste trabalho. Os resultados das simulações se mostraram compatíveis com as principais referências da literatura sobre as técnicas de elicitação, o que representou mais uma validação para o trabalho.

Para garantir o uso efetivo do modelo proposto neste trabalho, foi desenvolvido um portal corporativo que permite o cadastro de empresas e projetos de software, a configuração de cenários relativos aos projetos, a simulação desses cenários e a visualização dos resultados das simulações através de gráficos. Novas funcionalidades ainda estão sendo incorporadas ao portal, que, em breve, estará disponível através da internet.

Com isso, todos os objetivos específicos propostos para este trabalho também foram alcançados.

Através desse trabalho foi possível verificar que o uso de um modelo de dinâmica de sistemas que represente a atividade de elicitação de requisitos pode auxiliar na identificação de riscos envolvidos nas tomadas de decisão em projetos de software, favorecendo a adoção de medidas preventivas.

A partir do modelo apresentado, é possível verificar, por exemplo, como a experiência dos *stakeholders* envolvidos no projeto afeta a produtividade da elicitação, o quanto influencia o número de pessoas (elicitadores e *stakeholders*) no projeto, como as técnicas de elicitação impactam na produtividade da equipe e na qualidade dos requisitos, qual a relação de custo e produtividade, etc. Conhecendo-se essas relações, é possível evitar a tomada de decisões reativas, que não levam em consideração as variáveis dinâmicas envolvidas.

6.1 Principais contribuições

As principais contribuições disponibilizadas por este trabalho foram:

- a) Proposta de um modelo de dinâmica de sistemas para a atividade de elicitação do processo de engenharia de requisitos.
- b) Identificação e seleção das principais variáveis dinâmicas da atividade de elicitação de requisitos e descrição dos seus relacionamentos.
- c) Compilação de resultados de diversas pesquisas sobre comparações de técnicas de elicitação de requisitos.
- d) Disponibilização de um portal corporativo para uso efetivo de modelos de simulação por empresas de software.

Acredita-se que os resultados são relevantes, visto que apresentam uma nova possibilidade de auxílio para os gerentes de projetos de empresas de software.

6.2 Trabalhos futuros

A seguir, são apresentadas algumas sugestões de trabalhos futuros:

- a) **Construir um modelo de dinâmica de sistemas para a atividade de análise, no contexto da engenharia de requisitos.** Considerando que este trabalho modelou a atividade de elicitação de requisitos e que o trabalho de AMBRÓSIO (2008) criou modelos para as atividades de especificação e validação de requisitos, ficou uma lacuna para ser preenchida com um

modelo para a atividade de análise. Em um modelo para análise poderiam ser avaliadas situações como a classificação e priorização de requisitos, conflitos de requisitos e sua negociação, e como essas situações são influenciadas pelo número de *stakeholders* no projeto.

- b) **União dos modelos já construídos.** Após o desenvolvimento de um modelo para a atividade de análise, todo o contexto da engenharia de requisitos estaria modelado. O próximo passo seria então a união dos três modelos (modelo produzido por este trabalho, modelo de análise e o modelo de AMBRÓSIO (2008)). Para isso, algumas adaptações deveriam ser feitas para que os modelos trabalhassem com parâmetros compatíveis. Essa adaptação pode ser trabalhosa, visto que um exemplo de transformação necessária seria encontrar ou desenvolver uma relação de cláusulas por pontos de função, que foram as duas medidas adotadas para quantificar requisitos, neste trabalho e no trabalho de AMBRÓSIO, respectivamente.
- c) **Validar os modelos com dados reais de empresas.** Para garantir que os modelos produzidos estejam em conformidade com o que é visto na prática, é importante que eles sejam validados a partir de dados reais de projetos de software. Essa validação pode fazer com que os modelos sofram algum tipo de adaptação, como inclusão de novas variáveis, novos relacionamentos, calibração de valores, etc.
- d) **Atualização do Portal Corporativo.** Os novos modelos desenvolvidos poderão ser incluídos no portal. Além disso, podem ser incluídas novas funcionalidades, como a possibilidade dos gerentes construírem e adaptarem os seus próprios modelos, baseados nos já existentes.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDEL-HAMID, T. K.; MADNICK, S. E. **Software Project Dynamics: an Integrated Approach**. Englewood Cliffs: Prentice Hall, 1991. 264 p.
- AMBRÓSIO, B. G. **Modelagem da fase de requisitos em processos de desenvolvimento de software: uma abordagem utilizando dinâmica de sistemas**. Dissertação (Mestrado em Ciência da Computação), Universidade Federal de Viçosa, 2008.
- AMBRÓSIO, B. G.; BRAGA, J. L.; OLIVEIRA, A. P. Um modelo dinâmico para análise dos impactos da rotatividade de pessoal durante a fase de requisitos. In: XXII SBES - Simpósio Brasileiro de Engenharia de Software. **Proceedings...** Campinas, SP, 2008.
- ANDRIOLE, S. The politics of requirements management. **IEEE Software**. November–December. p. 82–84, 1998.
- ANTHONY, R. N. **Planning and Control Systems: A Framework for Analysis**. Harvard University, Boston, USA, 1965.
- AURUM, A.; WOHLIN, C. The fundamental nature of requirements engineering activities as a decision-making process. **Information and Software Technology**, 45 (14), p. 945–954, 2003.
- BASILI, V.R. Software modeling and measurement: The goal/question/metric paradigm. **Technical Report**, CS-TR-2956, Department of Computer Science, University of Maryland, College Park, MD 20742, September 1992.
- BOEHM, B. W. **Software Engineering Economics**, Englewood Cliffs, NJ: Prentice-Hall, 1981.
- BOEHM, B.; BASILI, V. R. Software Defect Reduction Top 10 List, **IEEE Computer**, v. 34, n. 1, p. 135-137, 2001.
- BRAGA, J. L.; SILVA, C. A. B.; WIAZOWSKI, B. A.; AVELLAR, S. O. C. Modelagem com dinâmica de sistemas. In: SANTOS, M. L.; VIEIRA, W. (Eds.). **Métodos Quantitativos em Economia**. Viçosa, MG: UFV, 2004. p. 411-434.
- BROOKS, F. P. **The Mythical Man-Month: Essays on Software Engineering**. 20th Anniversary Edition. Boston: Addison-Wesley, 1995. 322 p.
- BURTON, A.M.; SHADBOLT, N.R.; HEDGECOCK, A.P.; RUGG, G. A formal evaluation of knowledge elicitation techniques for expert systems: Domain 1. In

- D.S. Moralee, Ed., **Research and Development in Expert Systems IV**, pp. 136-145. Cambridge: Cambridge University Press.
- BURTON, A.M.; SHADBOLT, N. R.; RUGG, G.; HEDGECK, A.P. Knowledge elicitation techniques in classification domains. In Y. Kodratoff, Ed., **ECAI-88: Proceedings of the 8th European Conference in Artificial Intelligence**, pp. 85-90. London: Pitman.
- BURTON, A. M.; SHADBOLT, N. R.; RUGG, G.; HEDGECK, A. P. The efficacy of knowledge acquisition techniques: A comparison across domains and levels of expertise. **Knowledge Acquisition**, vol. 2, pp. 167-178, 1990.
- CHENG, B. H. C.; ATLEE, J. M. Research Directions in Requirements Engineering. In: **INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING**, 29., 2007, Minneapolis. **Proceedings...** Minneapolis: IEEE Computer Society, 2007. p. 285-303.
- CORBRIDGE, B.; RUGG, G.; MAJOR, N.P.; SHADBOLT, N.R.; BURTON, A.M. Laddering – technique and tool use in knowledge acquisition. **Knowledge Acquisition**, vol. 6, pp. 315-341, 1994.
- DAVIS, A. **Software Requirements - Analysis and Specification**. Prentice-Hall, UK, 1990.
- DAVIS, A.; DIESTE, O.; HICKEY, A.; JURISTO, N.; MORENO, A. M. Effectiveness of Requirements Elicitation Techniques: Empirical Results derived from a Systematic Review. **Proceedings of the 14th IEEE International Conference on Requirements Engineering**, Minneapolis, USA. Setembro, 2006, pp. 179-188.
- DIESTE, O.; JURISTO, N. Systematic Review and Aggregation of Empirical Studies on Elicitation Techniques. **IEEE Transactions on Software Engineering**, 11 Fev. 2010. IEEE computer Society Digital Library. IEEE Computer Society.
- DIESTE, O.; LOPEZ, M.; RAMOS, F. Updating Systematic Review about Selection of Software Requirements Elicitation Techniques. In Workshop on Engenharia de Requisitos (WER 2008). **Proceedings...** Barcelona, Espanha, 2008.
- DIESTE, O.; JURISTO, N.; SHULL, F. Understanding the Customer: What Do We Know about Requirements Elicitation. **IEEE Software**, vol. 25, p. 11-13, 2008.
- ERICSON, K.A.; SIMON, H.A. **Protocol Analysis: Verbal Reports as Data**. Cambridge, MASS: Bradford Books/ MIT Press, 1984.
- FORRESTER J.W. **Industrial Dynamics**. MIT Press: Cambridge, MA, 1961.
- FORRESTER J.W. **Principles of Systems**. MIT Press: Cambridge, MA, 1968.

- FOWLKES, J. E.; SALAS, E.; BAKER, D.P. The utility of event-based knowledge elicitation. **Human Factors**, vol. 45, pp. 24-35, 2000.
- GAMMACK, J. G. Different techniques and different aspects of declarative knowledge. In A. L. Didd, Ed. **Knowledge Acquisition for Expert Systems: A practical Handbook**, pp. 137-163, New York: Plenum Press, 1987.
- GOGUEN, J. A.; LINDE, C. Techniques for requirements elicitation. **Proceedings of IEEE International Symposium on Requirements Engineering**, pp. 152-164, 1993.
- GRIFFIN, A.; HAUSER, J. R. The voice of the customer. **Marketing Science** vol. 12, pp. 1-27, 1993.
- HICKEY, A.; DAVIS, A. Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes. **Proceedings of the Thirty-Sixth Annual Hawaii International Conf. on Systems Sciences (HICSS)**, IEEE CS, 2003.
- HÖST, M.; REGNELL, B.; NATT och DAG, J.; NEDSTAM, J.; NYBERG, C. Exploring bottlenecks in market-driven requirements management processes with discrete event simulation, **The Journal of Systems and Software**, Vol. 59, pp 323-332, 2001.
- IEEE – Institute of Electrical and Electronics Engineers. IEEE Recommended Practice for Software Requirements Specifications. IEEE Standard 830-1998.
- IEEE – Institute of Electrical and Electronics Engineers. IEEE Guide for Developing System Requirements Specifications. IEEE Standard 1233-1998.
- IEEE – Institute of Electrical and Electronics Engineers. IEEE Guide for Information Technology-System Definition-Concept of Operations. IEEE Standard 1362-1998.
- JOHNSON, L.; JOHNSON, N. E. Knowledge elicitation involving teachback interviewing. In: **Knowledge Acquisition for Expert Systems: A Practical Handbook**, ed. Kidd, A. L. New York, NY, USA: Plenum, 1987. pp. 91-108.
- KELLNER, M.I.; MADACHY, R.J.; RAFFO, D.M. Software process simulation modeling: Why? What? How?. **Journal of Systems and Software**. vol 46 (2/3), 1999, pp. 91-105.
- KORSON, T. D.; VAISHNAVI, V. K. Managing emerging software technologies: a technology transfer framework, **Communications of the ACM**, 35 (9), p. 101–113, 1992.
- KOTONYA, G.; SOMMERVILLE, I. **Requirements Engineering: process and techniques**. New York: John Wiley and Sons Ltd, 1998. 294 p.

- LAFRANCE, M. The knowledge acquisition grid: A method for training knowledge engineers. **Proceedings of the Knowledge Acquisition for Knowledge Based Systems Workshop**, Banff, AB, Canada, 1986.
- LAMSWEERDE, A. Requirements engineering in the year 00: a research perspective. In: **Proceeding of the IEEE International Conference on Software Engineering (ICSE)**, p. 5–19, 2000.
- LANE D.C.; ROUWETTE E.A.J.A.; VENNIX J.A.M. System dynamics in organizational consultation: modeling for interventions in organizations. **Special Edition of the international journal Systems Research and Behavioral Science**. 23(4): 443–573, 2006.
- LANE, D. C. The emergence of use of diagramming in system dynamics: A critical account. **Systems Research and Behavioral Science**. 25: 3-23, 2008.
- MADACHY, R. **Software Process Dynamics**. Wiley/IEEE Computer Society Press, 2007. 591 p.
- MCCONNELL, S. **Software Project Survival Guide**, Microsoft Press, Redmond, Washington. 1998.
- NUSEIBEH, B.; EASTERBROOK, S. Requirements engineering: a roadmap. In: International Conference on Software Engineering, 22, 2000, Limerick, Ireland. **Proceedings...** Limerick, Ireland: ACM, 2000. p. 35-46.
- PARNAS, D. L. Software engineering programmes are not computer science programmes. **Ann. Soft. Eng.**, 6(1):19–37, 1999.
- PETERSON, I. **Fatal defect: chasing killer computer bugs**. New York, NY: Vintage Books, 1996. 268 p.
- PFLEEGER, S. L. **Engenharia de software: teoria e prática**. 2.ed. São Paulo: Prentice Hall, 2004. 560p.
- PRESSMAN, R. **Software Engineering: A Practitioner's Approach**. 6.ed. New York: McGraw Hill, 2005. 752p.
- RAMESH, B.; JARKE, M. Towards Reference Models for Requirements Traceability. **IEEE Transactions on Software Engineering**, v. 27, n. 1, p. 58-93, 2001.
- RICHARDSON G, Problems with causal-loop diagrams. **System Dynamics Digest**, 1986.
- RICHMOND, B. M. **An Introduction to Systems Thinking**. Lebanon, NH: High Performance Systems, 2001. 173 p.
- ROBINSON, W. N.; PAWLOWSKI, S. D.; VOLKOV, V. Requirements Interaction Management. **ACM Computing Surveys**, Vol. 35, No. 2. p. 132-190, 2003.

- RODRIGUES, A. G.; WILLIAMS, T. M. System dynamics in software project management: towards the development of a formal integrated framework. **European Journal of Information Systems**, 6 (1), p. 51–66, 1997.
- SAYÃO, M.; LEITE, J. C. S. P. Rastreabilidade de Requisitos. **Revista de Informática Teórica e Aplicada**, v. 13, n. 1, p. 57-86, 2006.
- SENGE, P. **The Fifth Discipline: The Art and Practice of the Learning Organization**. New York: Currency Doubleday, 1990. 371 p.
- SIMMONS, D. B. Communications: a software group productivity dominator. **Software Engineering Journal**, v. 6, n. 6, p. 454-462, 1991.
- SOMMERVILLE, I. **Software Engineering**. 8.ed. Addison Wesley, 2007. 840p.
- SOMMERVILLE, I. SAWYER, P. **Requirements Engineering - A Good Practice Guide**. Lancaster University, Wiley, 1997.
- STERMAN, J. D. **Business dynamics: systems thinking and modeling for a complex world**. Boston: Irwin McGraw-Hill, 2000. 982 p.
- STRIGINI, L. Limiting the dangers of intuitive decision-making. **IEEE Software**. January, p. 101-103, 1996.
- SWEBOK, **Guide to the Software Engineering Body of Knowledge**. 2004 Version. A project of the IEEE Computer Society Professional Practices Committee. Disponível em: <<http://www.swebok.org/>>. Acesso em: 30 de março de 2010.
- VENSIM. **Vensim from Ventana Systems, Inc.** Disponível em: <<http://www.vensim.com>>. Acesso em: 19 de agosto de 2010.
- WIEGERS, K. E. **Software Requirements**. 2.ed. Redmond, Washington: Microsoft Press, 2003. 516p.