

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/242602092>

Estudo Comparativo sobre as Técnicas de Elicitação de Requisitos do Software

Article · January 2008

CITATIONS

9

READS

2,952

3 authors, including:



[Anderson Belgamo](#)

Universidade Metodista de Piracicaba (Unimep)

16 PUBLICATIONS 140 CITATIONS

[SEE PROFILE](#)



[Luiz Eduardo Galvão Martins](#)

Universidade Federal de São Paulo

55 PUBLICATIONS 289 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Software Engineering for Safety-Critical Systems [View project](#)

Estudo Comparativo sobre as Técnicas de Elicitação de Requisitos do Software

Anderson Belgamo
u9816034@fcti.unimep.br

Luiz Eduardo Galvão Martins (Orientador)
lgmartin@unimep.br

Universidade Metodista de Piracicaba - UNIMEP
Rod. do Açúcar, Km 156 13400-901
"Campus" Taquaral - Piracicaba-SP

Resumo

Este artigo tem como objetivo apresentar um estudo comparativo sobre as técnicas de elicitação de requisitos do software, sendo que a elicitação de requisitos é a primeira fase dentro da Engenharia de Requisitos. A elicitação de requisitos é de extrema importância para a construção do software, oferecendo significativas contribuições para a Engenharia de Software. Na elicitação de requisitos existem dois grupos de problemas: essenciais e acidentais. Para o enfrentamento desses problemas várias técnicas de elicitação de requisitos tem sido propostas, as quais estão sendo estudadas e avaliadas de forma a apontar as diferenças e semelhanças entre elas. Na avaliação das técnicas estudadas, propomos vários parâmetros de avaliação. Entendemos que estes parâmetros são importantes dentro do processo de avaliação das técnicas, pois são balizas úteis para a comparação entre as mesmas.

Abstract

The goal of this article is to show a comparative study about the software requirement elicitation techniques, where the requirements elicitation is the first step in the Requirements Engineering. The requirements elicitation is very important for software construction, offering meaningful contributions for Software Engineering. During requirements elicitation there are two problem groups: essential and accidental. To overcome these problems several requirement elicitation techniques have been proposed, which are being evaluated and analyzed to point out the differences and similarities among them. In evaluation of the techniques studied we propose a lot of parameters of evaluation. We understand that these parameters are important in the process of evaluation of the techniques, because they are useful references in the comparison among them.

Palavras-Chave

Requisitos, Elicitação de Requisitos, Engenharia de Requisitos

1. Introdução

Este artigo é resultante das atividades desenvolvidas no projeto de iniciação científica intitulado “Um Estudo Comparativo sobre as Principais Técnicas de Elicitação¹ de Requisitos do Software”. Este projeto, que ainda encontra-se em desenvolvimento, tem como objetivo estudar as principais técnicas de elicitação de requisitos do software, comparando-as de forma a explicitar pontos fortes e fracos de cada técnica. A elicitação de requisitos do software é uma atividade importante da Engenharia de Requisitos, que está inserida dentro do contexto da Engenharia de Software. A Engenharia de Requisitos se preocupa com a elicitação, especificação, análise, verificação e gerenciamento dos requisitos do software. A elicitação de requisitos oferece a base para as demais atividades da Engenharia de Requisitos.

2. Engenharia de Requisitos

2.1. Definição

A engenharia de requisitos, segundo (Thayer, 1997), é a primeira etapa dentro de todo o processo da engenharia de software, a qual estuda como coletar, entender, armazenar, verificar e gerenciar os requisitos. A principal preocupação na engenharia de requisitos é entender quais são os reais requisitos do sistema, bem como a documentação dos mesmos.

2.2. Fases da Engenharia de Requisitos

Como especificado em (Thayer, 1997) as fases da Engenharia de Requisitos são: elicitação, análise, especificação, verificação e gerenciamento. Elicitação: é o processo através do qual clientes e usuários são questionados por um desenvolvedor para falarem “o quê” o software deve fazer (ou seja, os requisitos). Esta fase será mais detalhada na seção 2.3. Análise: é o processo onde são analisadas as necessidades dos clientes e usuários para se chegar na definição dos requisitos de software. Especificação: é o processo de criação de um documento no qual estão definidos todos os requisitos analisados. Verificação: é o processo que busca assegurar que a especificação de requisitos de software está em concordância com os requisitos elicitados e analisados nas fases anteriores, ou seja, estão de acordo com o desejo do cliente. Gerenciamento: é o planejamento e controle da atividade de elicitação, especificação, análise e verificação dos requisitos.

2.3. A Elicitação de Requisitos do Software

A elicitação de requisitos é o início para toda a atividade de desenvolvimento de software, onde técnicas de elicitação são utilizadas. Embora a elicitação de requisitos seja a primeira atividade na engenharia de requisitos, esta atividade não acontece somente uma vez, seu processo é iterativo, ou seja, todas as demais etapas da engenharia de requisitos podem conter elicitação de requisitos. Para começarmos a elicitação precisamos identificar as pessoas certas, ou seja, os usuários finais. Ao iniciar a elicitação, os usuários podem ter um forte papel no processo de desenvolvimento, através de seus conhecimentos e experiências (Goguen, 1997). Para a escolha dos usuários precisamos nos preocupar que todas as informações dadas por eles sejam de confiança, ou seja, estas informações serão usadas para a fase de projeto, estando de acordo com a necessidade da empresa. Portanto, este usuário é de extrema

¹ A palavra elicitação vem do inglês *elicitation*, que significa descobrir algo obscuro.

importância, pois a escolha de uma pessoa errada acarretará em uma perda de tempo e dinheiro para ambas as partes envolvidas no desenvolvimento do software.

Como a natureza dos requisitos é mutante, ao iniciarmos a fase de elicitação de requisitos, os clientes podem mudar seus pensamentos, uma vez que eles vêem várias possibilidades mais claramente em momentos posteriores (Goguen, 1997), fazendo com que os requisitos mudem e, conseqüentemente, toda a fase de elicitação e análise dos requisitos sofrem alterações. Essas possibilidades podem vir de mudanças nos fatores sociais, financeiros, psicológicos e/ou políticos. Além da identificação dos usuários o desenvolvedor precisa conhecer o contexto no qual as informações estão situadas. Segundo (Jark, 1994) o contexto é dividido em 3 mundos: sujeito, uso e sistema. O sujeito representa uma parte do mundo exterior, onde o sistema – representado por alguma descrição estruturada – existe para servir algum indivíduo ou usuário (Siddiqi, 1997). Segundo Goguen, requisitos são informações e todas as informações estão situadas num contexto e isto determina o significado dos requisitos. Levar em consideração o contexto, significa prestar atenção em fatores sociais e técnicos. Requisitos emergem de situações sociais entre usuários e analistas (Siddiqi, 1997). Dessa interação com o usuário é que nascem os requisitos para a construção do software, porém, a parte mais difícil da elicitação de requisitos não é o ato de registrar o que os usuários querem; na verdade a atividade de elicitação é exploratória e desenvolvimental, ajudando os usuários a compreenderem melhor o que eles querem (McConnel, 1998). Uma antecipação do projeto, onde todos requisitos ainda não estejam elicitados, levaria a um erro, pois o software desenvolvido não estaria de acordo com as necessidades do cliente, acarretando gastos financeiros e de tempo.

2.4. Problemas encontrados durante a Elicitação de Requisitos do Software

Durante a elicitação de requisitos, e até mesmo no início da elicitação, aparecerão muitos problemas. No início da elicitação temos o problema da escolha dos usuários a serem entrevistados, pois uma escolha errada pode levar à perda de tempo e prejuízos financeiros para o desenvolvedor e para os clientes. Já durante a elicitação aparecerão problemas de escopo, de entendimento e de volatilidade. Além dos problemas citados acima teremos dois grandes grupos de problemas: problemas acidentais e problemas essenciais (Faulk, 1997).

a) Problemas Acidentais: São aqueles oriundos da falta de controle sobre aquilo que precisa ser construído, dentre os quais podemos destacar: pouco esforço despendido no levantamento de informações junto ao usuário; documentação pobre sobre os requisitos obtidos, pouca revisão dos requisitos obtidos; especificações incorretas dos requisitos e tendência em iniciar logo o processo de desenvolvimento do software (Martins, 1999), o que leva o usuário a não estar satisfeito com o resultado final do projeto. As dificuldades acidentais, segundo (Faulk, 1997), são: 1) Documentação escrita como uma reflexão tardia: isto continua sendo uma prática comum, onde a documentação de requisitos é desenvolvida depois que o software tenha sido escrito. Tal procedimento inevitavelmente viola o princípio da definição do “o que” o sistema deve fazer antes de “como”. 2) Não projetada para ser proveitosa: freqüentemente, na vontade implementar logo, pouco esforço é despendido para projetar, escrever, checar, ou gerenciar a administração da criação e evolução do documento de especificação. O resultado mais óbvio é uma organização pobre do software. O documento resultante não é uma referência técnica efetiva. A especificação é difícil para usar e difícil para manter.

b) Problemas Essenciais: São aqueles inerentes à elicitação de requisitos, dentre os quais podemos destacar: dificuldades do usuário em saber efetivamente o que ele quer, dificuldade de comunicação entre usuário e desenvolvedor e a natureza mutante dos requisitos (Martins,

1999). Segundo (Faulk, 1997) os problemas essenciais seriam causados por dois fatores primordiais: 1) Compreensão: pessoas não sabem o que elas querem. Isto não significa que as pessoas não tenham uma idéia geral do que o software fará. Ao invés disso, eles não começam com um preciso e detalhado entendimento de que funções pertencem ao software, o que as saídas devem fazer para cada possível entrada, quanto tempo cada operação deveria levar, como uma decisão afetará outra, e assim por diante. Muitas decisões ainda não foram tomadas, e expectativas mudarão na medida que o problema é melhor entendido. 2) Comunicação: requisitos de software são difíceis para se comunicar efetivamente. A dificuldade inerente da comunicação é composta pela diversidade de pessoas e audiências para a especificação de requisitos.

c) Diferenças: Podemos afirmar que os problemas acidentais são mais fáceis de se evitar, dependendo apenas da utilização das fases da engenharia de requisitos citadas na seção 1.2. Porém, os problemas essenciais envolvem a comunicação entre pessoas e esse processo deve levar em conta o contexto, as habilidades pessoais do entrevistado, o lado psicológico, ou seja, apenas uma abordagem tecnológica não poderá solucionar esses problemas, pois os aspectos sociais assumem grande importância na elicitação dos requisitos (Martins, 1999).

3. Técnicas de Elicitação de Requisitos

Para uma boa elicitação de requisitos, existem técnicas para o melhor entendimento e comunicação entre clientes e analistas, para que problemas, como os citados nas seções anteriores, não ocorram, ou se ocorrerem, que sejam mais facilmente resolvidos. Segundo (Jackson, 1995), uma técnica deve explorar as características específicas do problema e como as características do problema variam muito, nós precisamos de um repertório de métodos para cada classe de problema (Siddiqi, 1997). O problema da elicitação de requisitos não pode ser resolvido com uma abordagem puramente tecnológica, porque nesta fase o contexto social é mais crucial do que na fase de programação, especificação e projeto. A seguir será apresentado um elenco de técnicas que foram pesquisadas e que serão comparadas na tabela da seção 4.

a) Observação

A observação possibilita um contato pessoal e estreito do pesquisador com o fenômeno pesquisado, o que apresenta uma série de vantagens. (Damian, 1997) . As técnicas de observação são extremamente úteis para “descobrir” aspectos novos de um problema. Isto se torna crucial nas situações em que não existe uma base teórica sólida que oriente a coleta de dados. Ao mesmo tempo em que o contato direto e prolongado do pesquisador com a situação pesquisada apresenta as vantagens mencionadas, envolve também uma série de problemas. Algumas críticas são feitas ao método de observação, primeiramente por provocar alterações no ambiente ou no comportamento das pessoas observadas. Outra crítica é a de que este método se baseia muito na interpretação pessoal. Além disso há críticas no sentido de que o grande envolvimento do pesquisador leve a uma visão distorcida do fenômeno ou a uma representação parcial da realidade.

b) Entrevista

Entrevista é uma técnica de elicitação de requisitos muito usada. O engenheiro de requisitos ou analista discute o sistema com diferentes usuários , a partir da qual elabora um entendimento de seus requisitos. Há basicamente, segundo (Kotonya, 1998), 2 tipos de

entrevista: a) entrevistas fechadas onde o engenheiro de requisitos procura as perguntas para um conjunto um pré-definido de questões; b) entrevistas abertas onde não há agenda pré-definida e o engenheiro de requisitos discute, de modo aberto, o que os usuários querem do sistema. Entrevistas podem ser efetivas para desenvolver um entendimento do problema e para eliciar muitos requisitos gerais do sistema. Usuários finais são usualmente felizes para descreverem seus trabalhos e as dificuldades que eles enfrentam de forma relativamente natural, entretanto eles podem ter expectativas não realistas sobre o suporte que o computador dará. Portanto, entrevistas são muito menos efetivas para entendimento do domínio da aplicação e para o entendimento das questões organizacionais as quais afetam os requisitos.

c) Análise de Protocolo

A análise de protocolo pede à pessoa se engajar em alguma tarefa e correntemente falar sobre esta tarefa, explicando o seu pensamento do processo. Usuários alegam que esse tipo de linguagem pode ser considerada uma “verbalização direta do processo cognitivo específico”. A análise de protocolo não é um guia confiável sobre o que as pessoas estão pensando, ela está sujeita a problemas de interpretações pelos analistas. A restrição em estudar protocolos é que as pessoas podem produzir linguagens que oferece um perfil de atividade cognitiva autônoma. Segundo (Goguen, 1997), mesmo se fosse possível conseguir um perfil de atividade cognitiva autônoma da pessoa, tal objeto seria inapropriado para o processo de requisitos, porque o cliente não tem qualquer modelo mental pré-existente do sistema desejado. Os clientes têm conhecimento sobre negócios e necessidades organizacionais, enquanto o time de requisitos tem conhecimento sobre as possibilidades técnicas.

d) JAD²

JAD é uma marca registrada da IBM. O tema principal do JAD é colocar autoridades representativas e gerenciais juntas dentro de um *workshop* estruturado para promover decisões. Segundo (Damian, 1997) JAD consiste de 5 fases: definição do projeto, pesquisa, preparação para a sessão JAD, a sessão JAD, o documento final. As fases de definição de projeto e pesquisa no processo JAD lidam com a coleta de informações. A sessão JAD é então usada para validar as informações recolhidas nas fases anteriores. O processo JAD concentra-se na sessão JAD, e deste modo JAD contribui para a eliciação de requisitos como significado para validar as informações já colhidas. Na sessão JAD, as pessoas certas têm que estar envolvidas, e a presença de um facilitador pode ajudar a manter a sessão focalizada e minimizar ataques e defesas emocionais improdutivas. JAD promove cooperação, entendimento, e trabalho em grupo entre os vários grupos de usuários e o pessoal de sistemas de informação.

e) PD³

Tradicionalmente valores democráticos, os quais tem sido levados em conta no processo de projeto, tem sido somente aqueles concentrados em fatores técnicos e econômicos. Mas com o uso do PD fatores técnicos-sociais tem sido levados em conta. O projeto deveria ser feito com o usuário. Aprendizado mútuo deveria ser uma parte importante

² *Joint Application Development*

³ *Participatory Design*

do trabalho em um grupo de projeto, não sendo meramente uma visita aos laboratórios de pesquisa (Damian, 1997). Trabalhadores e clientes são inteligentes e criativos, contribuidores produtivos para as organizações, desde que sejam encorajados a expressarem seus desejos, aplicarem sua esperteza e exercitarem suas capacidades de tomar decisões, assumindo responsabilidades do impacto de suas ações.

f) QFD⁴

O termo qualidade é definido como “um conjunto de meios para produzir economicamente produtos ou serviços, os quais satisfaçam os requisitos do cliente”. QFD é “um conceito que provê meios de interpretar requisitos do cliente em requisitos técnicos, apropriados para cada estágio do desenvolvimento e produção do produto” (Damian, 1997). As fases iniciais do QFD podem ser descritas como sendo “simplesmente um sistema de identificação e priorização das necessidades do cliente obtidas de cada recurso avaliado”. QFD é um conceito que aplica-se bem para a eliciação de requisitos, especialmente num modelo de eliciação onde a voz do cliente é o guia para a criação de requisitos.

g) CRC⁵

Como definido em (Zhu, 1998), CRC é uma sessão de grupo, que é similar ao JAD, onde os papéis dos participantes e o papel do facilitador são claramente definidos. Em CRC, participantes consistem não somente de usuários e facilitador, mas também de outras pessoas envolvidas indiretamente no sistema. CRC é diferente de JAD e QFD pois ele foca-se no usuário operativo.

h) Prototipação

Este método para eliciação de requisitos utiliza-se do uso de uma técnica de prototipação para a avaliação do usuário. O conjunto inicial de requisitos é usado como base para criar o Protótipo de Interface do Usuário com o sistema inicial (simplificado). Para essa criação, o projetista precisa manter o protótipo tão simples quanto possível. O ponto forte desta atividade é apresentar muitas alternativas para o usuário antes de se gastar muito esforço para qualquer protótipo em particular. Após a aceitação do protótipo pelos usuários, os desenvolvedores precisam criar um documento de especificação dos requisitos paralelo ao protótipo de interface (McConnel, 1998).

i) Cenários

Usuários finais e outros agentes do sistema acham a utilização de cenários mais fáceis para relacionar-se aos exemplos da vida real do que descrições abstratas das funções realizadas pelo sistema. Por essa razão, é freqüentemente útil desenvolver um conjunto de interação dos cenários, e usar estes para eliciar e clarear requisitos de sistema. Cenários são exemplos de sessões de interação as quais são concentradas com um tipo único de interação entre um usuário final e o sistema. Usuários finais simulam suas interações usando cenários. Eles explicam para o time de engenheiros de requisito o que eles estão fazendo e a informação da qual eles precisam do sistema para descrever a tarefa descrita no cenário.

⁴ *Quality Function Deployment*

⁵ *Cooperative Requirements Capture*

4. Parâmetros de Avaliação das Técnicas de Elicitação de Requisitos

A partir das técnicas citadas na seção 3, foram propostos vários parâmetros para a avaliação das técnicas (*vide tabela 1*). Os parâmetros propostos são explicados abaixo:

a) Grupo/Indivíduo: indica se a técnica é aplicada em grupo ou individualmente; b) Contexto: indica se a técnica leva em conta o ambiente onde está se realizando a elicitação; c) Caráter de interação: indica se o entrevistador e entrevistado sentem-se a vontade, num clima de estímulo e de aceitação mútua; d) Usa lado introspectivo: voltar-se a si próprio e pensar como seria o serviço; e) Confiabilidade: se as informações colhidas são confiáveis para o desenvolvimento do projeto; f) Custo: esforço gasto no uso da técnica; g) Qualidade: é um critério de validação onde é perguntado se na técnica há democracia, aprendizado mútuo, educação e resolução de conflito; h) Padronização: se a técnica possui uma regra para seu uso; i) Produtividade: se é uma técnica produtiva; j) Quantidade: é um critério de validação onde é perguntado se na técnica há índices de performance e economia de tempo; k) Descrição de ações dos usuários: se a técnica registra as gesticulações e faces do rosto do usuário; l) Compartilhamento de informações: se todos os indivíduos do grupo compartilham as informações; m) Tempo: tempo despendido para a elicitação de requisitos; n) Promover cooperação: se a técnica promove a cooperação entre os indivíduos do grupo; o) Facilitador: se possui uma pessoa com a função de guiar, levantar questões e discussões num grupo; p) Validar requisitos com os usuários: se a técnica valida seus requisitos com os usuários; q) Conflitos entre usuários do grupo: se na técnica existe um meio para lidar com conflitos em grupo; r) Atividade prematura de projeto: se a técnica evita que os analistas pensem que todos os requisitos já foram elicitados e que o projetista já possa começar a elaboração do sistema.

Tabela 1 - Cruzamento entre as técnicas de elicitação com os parâmetros de avaliação propostos

Técnicas de Elicitação	Parâmetros de Avaliação								
	Observação	Entrevista	Análise de Protocolo	JAD	PD	QFD	CRC	Proto-tipo-ção	Cenários
Grupo/Indivíduo	Grupo/Indivíduo	Indivíduo	Indivíduo	Grupo	Grupo	Grupo	Grupo	Grupo/Indivíduo	Grupo/Indivíduo
Considera o contexto	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Caráter de Interação	Não	Sim	Não	Sim	Sim	Sim	Sim	Sim	Sim
Liberdade de Percorso	Sim	Sim	Não	Não	Não	Não	Não	Não	Sim
Usa lado introspectivo	Sim	Sim	Sim	Não	Não	Não	Não	Não	Não
Confiabilidade	Média	Alta	Baixa	Alta	Alta	Alta	Alta	Alta	Alta
Custo	Baixo	Médio	Médio	Alto	Alto	Alto	Alto	Alto	Alto
Qualidade	Média	Média	Baixa	Alta	Alta	Alta	Alta	Alta	Alta
Padronização	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Produtividade	Baixa	Média	Baixa	Alta	Alta	Média	Alta	Alta	Alta
Quantidade	Baixa	Alta	Baixa	Alta	Média	Média	Média	Média	Média
Descreve ações dos usuários	Sim	Não	Não	Não	Não	Não	Não	Não	Sim
Compartilhamento de informações	Não	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim
Tempo	Longo	Médio	Médio	Longo	Longo	Longo	Longo	Médio	Longo
Promove cooperação	Não	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim

Facilitador	Não	Não	Não	Sim	Sim	Não	Sim	Não	Não
Valida requisitos com os usuários	Não	Sim	Não	Sim	Sim	Sim	Sim	Sim	Sim
Conflitos entre os usuários do grupo	Não	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim
Evita atividade de projeto prematura	Não	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim

5. Conclusão

O projeto exposto deverá resultar em uma monografia sobre o assunto, que auxiliará no desenvolvimento do ensino de engenharia de software em disciplinas do curso de Ciência da Computação. Os parâmetros de avaliação das técnicas foram obtidos estudando e analisando as diferentes técnicas de elicitação encontradas nos artigos, livros e páginas na *Internet*. Durante o levantamento bibliográfico sobre o tema não foi encontrado nenhum material propondo parâmetros de comparações entre as técnicas, reforçando a relevância do estudo comparativo das técnicas de elicitação de requisitos.

Um estudo de caso está sendo realizado e servirá como uma forma de aplicação das técnicas, que também auxiliará no refinamento dos parâmetros de avaliação propostos. O motivo desse estudo de caso é realizar uma avaliação efetiva entre as técnicas, podendo diferenciá-las, obtendo dados para avaliar onde cada técnica seja melhor aplicada, de maneira a ajudar o engenheiro de requisitos a escolher a técnica mais adequada para a situação defrontada. Durante o estudo de caso já foram aplicadas as técnicas de observação, questionário, análise de protocolo, JAD e entrevista. Após a aplicação das demais técnicas poderá ser feita uma melhor avaliação de seu uso, oferecendo subsídios para o engenheiro de requisitos escolher a técnica de elicitação mais adequada para a situação defrontada.

6. Referências Bibliográficas

- DAMIAN, Adrian, et al. “Joint Application Development and Participatory Design” 1997. <<http://www.cpsc.ucalgary.ca/~pand/seng/613/report.html>>
- FAULK, Stuart R. “Software Requirements: A Tutorial” in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p. 128-149
- GOGUEN, Joseph A. “Techniques for Requirements Elicitation” in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p.110 –122
- HARWELL, Richard. “What Is a Requirement” in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p. 23-29
- JACKSON, Michael. “Software Requirements and Specifications”, Addison-Wesley, Reading, Mass., 1995.
- JARK, M e POHL, K. “Requirements Engineering in 2001: (Virtualy) Managing a Changing Reality”, Software Requirements Engineering, Nov. 1994, p.p. 257-266.
- KOTONYA, Gerald e SOMMERVILLE, Ian. Requirements Engineering Processes e Techniques. John Wiley and Sons, 1998
- MARTINS, Luiz E. G. “Utilização dos Preceitos da Teoria da Atividade na Elicitação de Requisitos do Software”, XII Simpósio Brasileiro de Engenharia de Software, Out. 1999.
- McCONNEL, Steve. “Software Project Survival Guide: How to Be Sure Your First Important Project isn’t Your Last”. 1998, p.p 113-124.
- SIDDIQI, Jawed. “Requirements Engineering: the emerging wisdom” in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p. 36-40.
- THAYER, R. H. e DORFMAN, M.; “Introduction to Tutorial Software Requirements Engineering” in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p. 1-2.
- ZHU, David. “Requirements Engineering”. 1998. <<http://www.cpsc.ucalgary.ca/~zhud/seng693.html>>