A Importância do Uso das Técnicas de Extração de Requisitos para o Desenvolvimento de *Softwares*

Débora Carolina de Souza Rodrigues
Acadêmica do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, das Faculdades Integradas de Três Lagoas

Prof.ª M. Sc. Elisangela Citro
Coordenadora do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, das Faculdades Integradas de Três Lagoas

Resumo

Para grandes empresas o *software* se tornou indispensável, tanto pela velocidade com que se pode realizar tarefas, quanto pela redução de mão-de-obra. Como todo produto, o *software* também passa por um longo processo de produção, e um dos pontos primordiais para o desenvolvimento de um *software* é a extração de requisitos; a extração de requisitos é a etapa em que o desenvolvedor buscará saber o que o *software* em questão deverá realizar. Durante a extração de requisitos, o desenvolvedor poderá encontrar dificuldades pequenas, fáceis de serem resolvidas, como também encontrará dificuldades extremamente complexas. Por haver esses obstáculos, foram criadas técnicas para serem aplicadas na extração de requisitos, o objetivo dessas técnicas é facilitar ao desenvolvedor o trabalho de extração de requisitos. O objetivo deste artigo é, além de apresentar as técnicas, servir de apoio a estudos futuros de aplicação da extração de requisitos.

Palavra chave: extração de requisitos, análise, desenvolvedor, técnicas de extração de requisitos.

1. A Importância do *Software* para a Sociedade

O sistema de *software* possui importante papel perante a sociedade, desde facilitar a comunicação entre uma máquina e o homem, até controlar grandes movimentações de dinheiro entre países em lados opostos do mundo.

Quando se cria um *software*, o objetivo do mesmo é facilitar a vida das pessoas, seja lidando com máquinas, fazendo cálculos ou até mesmo em um simples editar de imagens.

O *software* é o motor que dirige a tomada de decisão nos negócios e no cotidiano das pessoas, seja como usuário do mesmo ou pelo beneficio que esse uso pode proporcionar. Serve de base à moderna investigação cientifica e às soluções de problemas de engenharia. Está embutido em sistemas de todas as naturezas: de transportes, médicos, de telecomunicações, militares, de processos industriais, de produtos de escritório, comerciais etc. O *software* é virtualmente inevitável no mundo moderno. Ele é o produto, e ao mesmo tempo, o veículo para entrega do produto.

Quer seja encontrado em um telefone celular, quer opere em um computador de grande porte, o *software* é um transformador de informação – produzindo, gerando, adquirindo, modificando, exibindo, ou transmitindo informações a todo instante para milhares de pessoas no mundo todo. Um exemplo claro desta transmissão de informações está nas bolsas de valores, em que a todo momento surge uma nova informação preciosa a muitos usuários.

O *software* entrega o mais importante produto da nossa época – a informação. O desenvolvimento de *softwares* tem crescido nos últimos anos devido a sua grande importância na sociedade moderna. O uso cada vez maior de computadores pessoais e nas diversas áreas do conhecimento humano tem gerado uma crescente busca por soluções que automatizem os diversos processos.

Vive-se em uma época onde "o capitalismo faz o mundo girar", e as empresas necessitam aumentar seus níveis de produção; consequentemente estas empresas irão aumentar seu número de funcionários, necessitarão realizar números cada vez maiores de cálculos, e muitas outras coisas. São nestas horas que os *softwares* se tornam indispensáveis, onde, fazendo uso do mesmo, é possível reduzir custos, aumentar níveis de produção, facilitar a organização, e agilizar processos a serem realizados dentro destas empresas.

Apesar do enorme avanço do desenvolvimento de *software* nos últimos anos, algumas empresas estão presas a antigos preconceitos, o que impede seu amadurecimento no processo de desenvolvimento. Elas não percebem que seus ambientes de trabalho estão cada vez mais complexos, o que torna cada vez mais desigual a relação entre eficiência e velocidade, problema que o *software* pode facilmente solucionar, pois com o avanço das tecnologias e a necessidade maior de informação, o produto de *software* tornou-se essencial e esta necessidade o fez evoluir em relação à segurança e confiabilidade em sua perfeita execução.

Não há, na sociedade moderna, algum ser humano que se diga imune ao domínio que o *software* tem sobre sua vida, seja ela educacional, profissional, pessoal, familiar ou sobre seus relacionamentos, pois os produtos de *software* estão presentes em equipamentos de produção industrial, transportes e logísticas, comércio, atendimentos básicos como saúde e educação, lazer e comunicação entre vários outros que seria impossível listar todos.

O crescimento da *internet* contribuiu para que este cenário mundial fosse cada vez mais dependente de sistemas informatizados, o que gerou uma explosão de *sites* comercias que se utilizam dos serviços *on-line* para proporcionar aos seus clientes conforto e praticidade. Hoje em dia, bancos e associações comerciais fazem uso ilimitado dessas tecnologias, como serviços *e-commerce* e *internet banking*, que são gerenciados por

excelentes sistemas de *softwares* e utilizados por milhares de pessoas que se sentiriam praticamente órfãs sem esse tipo de serviço.

Em plena era digital, o computador se tornou um bem comum, que tanto pode ser utilizados nas grandes empresas, como por qualquer pessoa, seja em sua casa, na escola ou em *lan houses*. Pessoas comuns como estudantes, donas de casa, crianças passam horas em frente a um computador, usufruindo de programas como MSN, *Photoshop*, *games*, e outros. Muitas vezes estas pessoas não fazem ideia de como é criado todos esses *softwares*, pois a elas só interessam as determinadas funções que eles irão realizar. O desenvolvedor ao criar um *software*, tem a função de levar a estas pessoas, facilidades para o dia a dia. Assim se produz um determinado programa, com seu manual e o entrega as pessoas que irão utilizá-lo conforme o que está explicito em seu manual.

Com base em todos estes fatos relatados, pode-se perceber como é grande a interação entre *software* e sociedade. Desta forma se torna cada vez maior a necessidade de um trabalho bem executado por parte dos desenvolvedores, que são os responsáveis diretos por levar até a sociedade todos os benefícios trazidos pelos *softwares*. Torna-se praticamente impossível descrever a sociedade atual sem falar em sistemas de *softwares* ou sistemas computacionais; eles estão ligados a economia mundial, a saúde, ao lazer, ao comércio, aos estudos, e tantas outras coisas do dia a dia das pessoas em todo o planeta.

2. A Extração de Requisitos

Os sistemas de *software* estão cada vez mais presentes no dia a dia das pessoas, e o bom funcionamento destes faz total diferença no desempenho de tarefas essenciais ao cotidiano, trazendo confiança a seus usuários. E para isso devem cumprir integralmente seus requisitos. Entretanto, a construção de sistemas para tal aplicação é complexa, pois deve lidar com requisitos intransigentes, restrições de integridade e a necessidade de um vasto conhecimento sobre a aplicação para que as interações entre o *software* e o ambiente possam ser adequadamente descritas. [CARVALHO, 2001]

As atividades executadas no início do desenvolvimento são de fundamental importância para todo o processo de construção do *software*, pois quando seu desenvolvimento é atribuído a um grupo, a comunicação e as informações adequadas são essenciais. Carvalho [2001] afirma que quando as atividades são executadas paralelamente, a sincronização e o controle dessas atividades passam a ser a chave para o sucesso.

A extração de requisitos refere-se ao ato de buscar todas as informações necessárias para o desenvolvimento de um *software* a fim de atender as necessidades do usuário. Para realizar essa tarefa o desenvolvedor realizará o uso de diversas técnicas como entrevistas, reuniões entre desenvolvedores e usuários, análises e pesquisas.

É essencial que a extração de requisitos seja feita com muita seriedade e competência, pois será responsável pelo sucesso do programa a ser desenvolvido. O desenvolvedor terá muito que trabalhar, pois são inúmeras as dificuldades encontradas no momento de extrair os requisitos necessários para o desenvolvimento do *software*. Quando os requisitos não são completamente entendidos, registrados e comunicados, [PRESSMAN, 1995], para a equipe de desenvolvimento, pode haver divergência entre o que o sistema a ser construído faz e o que deveria fazer.

A análise de requisitos é um processo de refinamento e especificação de um objetivo inicial. O objetivo do *software*, inicialmente estabelecido pelo engenheiro de *software* é refinado durante o projeto do sistema e aperfeiçoado em detalhes.

Segundo Carvalho [2001], existem várias técnicas para facilitar o entendimento entre desenvolvedor e usuários. Entre estas técnicas pode-se citar a entrevista. A entrevista consiste

em uma série de reuniões entre desenvolvedor e usuário. Estas reuniões devem ser marcadas pelo desenvolvedor e informadas com antecedência ao usuário.

O objetivo das entrevistas é fazer com que o cliente passe o máximo de informações possíveis ao desenvolvedor, sobre os requisitos necessários para a produção do *softwar*e em questão.

A análise de requisitos pode parecer uma tarefa fácil, mas o conteúdo da comunicação entre o desenvolvedor e o cliente desempenha um papel importante na análise e especificação de requisitos, e isso torna a interpretação mais suscetível a erros e o levantamento de informações falsas é muito provável, também como a ambiguidade dessas informações. Estes erros podem ocorrer devido aos ruídos, [PRESSMAN, 1995], durante a comunicação feita nas entrevistas entre usuários e desenvolvedores, como interpretação errada ou omissão de pontos importantes.

Conforme Pressman [1995], as duas primeiras fases da extração de requisitos, ou seja, o reconhecimento do problema e a síntese de avaliação e solução se baseiam na aquisição bem sucedida de informações. Frequentemente as informações fornecidas entram em conflito com outras exigências, ou a função e o desempenho do *software* é prejudicado por restrições impostas e a percepção dos objetivos tende a se modificar com o tempo.

Pode-se observar que "à medida que o tamanho do problema cresce a complexidade da análise também ganha outras proporções. Cada novo item de informação, restrição ou função tem efeito direto sobre todos os demais elementos", [PRESSMAN, 1995]. Há também a necessidade de eliminar inconsistências, detectar omissões e fazer com que um grande problema possa ser visualizado de todas as perspectivas possíveis, para que o mesmo se torne administrável.

As mudanças também devem ser tratadas de maneira cautelosa, pois não importa a complexidade do *software* ou sistema e nem o estágio em que se encontra seu desenvolvimento, ele se modificará. Neste momento as mudanças deverão coordenar com as exigências do *software*, deverá ser avaliado o impacto sobre outras partes do *software* aparentemente não relacionadas e erros de especificação deverão ser corrigidos, para que não haja efeitos indesejáveis.

Todos os problemas apresentados podem ter causas atribuídas, como uma comunicação falha, técnicas e ferramentas inadequadas que resultam em especificação inadequada, tendências para seguir atalhos durante a análise de requisitos, o que pode levar a um projeto instável, e a falta de se levar em consideração alternativas, antes que o *software* seja especificado.

Os problemas mencionados poderiam ser facilmente resolvidos com a aplicação de técnicas de comunicação sólidas, princípios de análise fundamentais e métodos de análise sistemáticos. [SOMMERVILLE, 2003].

Pode-se perceber como é trabalhoso e difícil o processo de extração de requisitos. O desenvolvedor deverá utilizar de muito conhecimento e sabedoria para contornar todas as dificuldades no processo de extração dos requisitos necessários para a produção de um produto de *software*.

O objetivo de um *software* é sempre fornecer facilidades à vida de empresas e pessoas, e um dos passos para a produção desses *softwares* é a extração de requisitos onde, mesmo sendo uma etapa rigorosamente sistemática, o desenvolvedor deve se empenhar ao máximo para resolver problemas relacionados à obtenção destes requisitos, sejam problemas técnicos, ou de comunicação entre desenvolvedor e usuário.

Em suma, para Carvalho [2001], têm-se como as maiores dificuldades na extração de requisitos:

• As atividades não podem ser totalmente separadas e executadas linearmente;

• As necessidades do usuário mudam à medida que o ambiente no qual o sistema funcional muda;

• Mudanças dos requisitos acontecem na maioria dos sistemas complexos;

• Falta de conhecimento do usuário das suas reais necessidades e do que o produto de *software* pode lhe oferecer;

• Falta de conhecimento do desenvolvedor do domínio do problema;

• Domínio do processo de extração de requisitos pelos desenvolvedores de *software*;

• Comunicação inadequada entre desenvolvedores e usuários;

• Dificuldade de o usuário tomar decisões;

• Problemas de comportamento;

• Questões técnicas.

3. Técnicas para a Extração de Requisitos

Extração de requisitos refere-se ao ato de buscar todas as informações necessárias para o desenvolvimento de um *software* a fim de atender as necessidades do usuário. É o processo de transformação das idéias do usuário em um documento formal; pode-se entender também como o processo de extrair os serviços que o cliente requer do sistema e as restrições sob as quais o sistema deve operar e ser desenvolvido. Para atingir tal objetivo, o desenvolvedor realizará o uso de diversas técnicas como entrevistas, reuniões entre desenvolvedores e usuários, análises e pesquisas.

Carvalho [2001] afirma que "as técnicas de extração e análise de requisitos visam superar as várias dificuldades inerentes ao processo. Algumas tratam das dificuldades de comunicação, enquanto outras tratam de dificuldades técnicas ou de comportamento humano. Algumas são de alto nível no sentido de que são técnicas amplas para o processo de extração e requisitos; outras são de baixo nível, pois fornecem táticas específicas para a extração de detalhes sobre uma determinada parte do produto ou um usuário específico."

Independente de quais sejam as técnicas escolhidas, é essencial que a extração de requisitos seja feita com muita seriedade e competência, pois esta será a responsável pelo sucesso do programa a ser desenvolvido. O desenvolvedor terá muito que trabalhar, pois são inúmeras as dificuldades encontradas no momento de extrair os requisitos necessários para o desenvolvimento do *software*.

Os problemas a serem resolvidos pelo desenvolvedor no momento em que se realiza a extração de requisitos podem ser simples e até acabar sendo ignorados, ou mais complexos e difíceis de serem resolvidos. Por esse motivo utiliza-se das várias técnicas de extração de requisitos para facilitar o entendimento entre desenvolvedor e cliente/usuário.

Algumas destas técnicas são genéricas, que poderão facilmente ser aplicadas no início da extração, e a partir de seus resultados, excluir aquilo que não será utilizado na análise de requisitos e também determinar qual será a técnica que mais se adaptará às necessidades do cliente/usuário. Pode-se apontar como técnicas genéricas perguntar a pessoas apropriadas; inferir as necessidades do *software* a partir da observação de seus prováveis usuários; discutir as necessidades e formular uma opinião comum; identificar problemas para verificar quais requisitos serão necessários e a suposição de características no caso de não haver usuários disponíveis.

3.1 - A Técnica Entrevista

Entre as técnicas específicas, pode-se citar a entrevista. A entrevista consiste em uma série de reuniões entre desenvolvedor e usuário. Estas reuniões devem ser marcadas pelo desenvolvedor e informadas com antecedência ao usuário.

O objetivo das entrevistas é fazer com que o cliente passe o máximo de informações possíveis ao desenvolvedor, sobre os requisitos necessários para a produção do *softwar*e em questão. Sendo assim, deve-se informar ao cliente, com antecedência, os objetivos da entrevista e lhe fornecer material necessário para que ele possa se preparar de acordo com as necessidades.

O desenvolvedor tem a função de fazer com que a entrevista venha a ser o mais construtiva possível. Não se trata de uma entrevista qualquer, tem que ser totalmente planejada e objetiva, pois dela será retirado tudo o que é necessário para a produção de um *software* que atenda as reais necessidades do usuário final.

3.2 - A Técnica *Brainstorming*

Pode-se destacar outra técnica chamada *brainstorming*. Esta técnica tem o mesmo objetivo da entrevista, que é facilitar o entendimento entre desenvolvedor e usuário, sempre visando à qualidade na produção do *software*.

Brainstorming é uma técnica utilizada basicamente para a geração de ideias, consiste em realizar reuniões envolvendo desenvolvedores e usuários que estão ligados diretamente ao produto que está sendo desenvolvido, onde se permite que as pessoas sugiram e explorem ideias sem que sejam criticadas ou julgadas. Nestas reuniões serão elaboradas ideias que posteriormente são analisadas e, se adequadas, consolidadas.

As reuniões devem ser dirigidas por um líder-desenvolvedor, que terá a função de dar início a reunião e buscar o envolvimento dos participantes, para que esta seja produtiva. Ele deverá fazer com que todas as ideias sejam ponderadas, e também dar caminho à reunião, de forma que fique bem claro a todos os objetivos que estão a ser discutidos. Esta técnica possui duas etapas, a primeira, de geração de ideias, é onde os participantes deverão expor suas ideias sem que haja discussão ou críticas em relação a elas, após o término da etapa de geração, inicia-se a etapa de consolidação, onde as ideias geradas serão filtradas e finalmente discutidas.

Ao final de tudo as ideias são consolidadas, e tornam-se requisitos apropriados para uma versão subsequente do produto de *software*. Quando se utiliza desta técnica, por ser um processo relativamente não estruturado, pode se ter menor qualidade ou baixo nível de detalhamento em relação a outros processos.

3.3 – A Técnica JAD

Outra técnica a ser considerada, é a *Joint Application Design*, ou JAD. Onde o trabalho em equipe, envolvendo pessoas empenhadas, com um bom líder, tem tudo para terminar em excelentes resultados. Esta é a ideia em que consiste a técnica denominada JAD.

A JAD promove a interação em equipe por parte de desenvolvedores e usuários. Todos devem estar envolvidos no trabalho a ser realizado, compartilhando uma visão geral do que o produto de *software* deve ser. Os desenvolvedores nem sempre saberão quais são os requisitos necessários para a produção de um determinado produto de *software*, e assim torna-se fundamental a participação de usuários, que fornecerão todos estes requisitos para que os desenvolvedores cheguem à conclusão do que será necessário para o desenvolvimento de um excelente produto de *software*.

Nesta técnica o líder tem uma responsabilidade imensa, pois ele é o responsável direto pelo sucesso do projeto. Como líder ele deve fazer com que a equipe trabalhe de forma organizada e objetiva, sempre com foco na excelência do produto em questão.

O desenvolvedor deve guiar todo o trabalho em equipe, motivando os integrantes, ao usar toda a sua capacidade de liderar, com sabedoria e sempre buscando maior conhecimento da área, e assim facilitar a dinâmica de grupo. Isso porque um dos principais fundamentos da JAD é a dinâmica, que torna o trabalho de extração de requisitos mais leve e traz o usuário para perto do problema em questão; com o mesmo objetivo, os recursos audiovisuais são para despertar a criatividade e o senso de imaginação dos participantes.

Dessa forma, a técnica JAD proporciona maior integração dos usuários e assim estrutura o debate, focando os requisitos necessários para a produção do *software.* O uso de documentação padrão, que é preenchida e assinada por todos os participantes da sessão, deixa tudo o que foi realizado devidamente registrado, facilitando seu entendimento.

3.4 – A Técnica PIECES

Há também a técnica PIECES, que segundo Carvalho [2001], fornece um conjunto de categorias de problemas que podem ajudar o analista a estruturar o processo de extração de requisitos. PIECES é uma sigla para seis categorias de questões a serem levadas em consideração: desempenho (*performance*), informação e dados, economia, controle, eficiência e serviços.

Esta técnica ajuda desenvolvedores inexperientes a dar início as entrevistas, trazendo de maneira estruturada todos os problemas que devem ser discutidos com o cliente/usuário durante o processo de extração de requisitos. Como na entrevista, a técnica PIECES ajuda a lidar com dificuldades de articulação do problema e comunicação.

3.5 – A Técnica Prototipagem

Prototipagem é outra técnica utilizada na extração de requisitos, onde se utiliza um *software* já existente como referência, voltada para aqueles clientes que tem maior dificuldade de expressar suas necessidades. Esta técnica consiste em utilizar um produto de *software* como modelo para maior entendimento do usuário sobre as características e benefícios que seu *software* poderá possuir. Caso não haja um produto de *software* que possa ser comparado ao produto final desejado pelo desenvolvedor e cliente, pode se usar a prototipagem para criar um produto com características semelhante às do produto desejado, desde que o mesmo seja feito de forma rápida sem interferir no processo de obtenção do produto final.

O protótipo só é uma boa escolha se não vier a ser utilizado como produto final, pois não passa de um esqueleto que esboça por meio de janelas e descrição de funções, aquilo para que o produto de *software* será desenvolvido para executar.

Quando um produto final já está elaborado, pode-se então descartar o protótipo utilizado.

4. Conclusão

A extração de requisitos é importantíssima para o desenvolvimento de um bom produto de *software,* pois “para projetar *software* de boa qualidade, é necessário compreender como ele será utilizado e como sofrerá alterações ao longo do tempo”, [PETERS, 2001]. Segundo Peters [2001], que afirma que “o grau de compreensibilidade, precisão e rigor da descrição fornecida por um documento de requisitos de *software* tende a ser diretamente proporcional ao grau de qualidade do produto resultante”, caso o desenvolvedor não consiga

contornar os problemas encontrados na extração dos requisitos, ou faça uma extração fraca em recursos e técnicas, o *software* produzido terá enorme quantidade de erros, e será um produto de baixíssima qualidade.

"A análise de requisitos é o principal processo executor em um sistema de *feedback* que produz descrições comportamentais e não-comportamentais do *software*", [PETERS, 2001], para auxiliar esse processo tão importante foram criadas técnicas que são aplicadas na extração de requisitos, o desenvolvedor terá a seu dispor o uso dessas técnicas e caberá a ele fazer com que a aplicação das mesmas renda bons resultados, determinando a qualidade de um produto de *software*.

Nenhuma técnica por si só será suficiente para o desenvolvimento de um projeto real. O desenvolvedor deve ser capaz de escolher um conjunto de técnicas que melhor se adaptem ao produto a ser desenvolvido [CARVALHO, 2001]. São várias as técnicas para extração de requisitos, e o desenvolvedor fará uso do conjunto que mais lhe parecer eficiente e prático. Todas as técnicas estão à disposição do desenvolvedor, e somente ele pode realizá-las com sucesso. Tudo vai depender da interação entre desenvolvedor e cliente/usuário.

Espera-se que os estudos aqui levantados sirvam de apoio a demais estudos referentes à área de Engenharia de *Software*, em particular, a área de Engenharia de Requisitos. Com base neste, pretende-se elaborar trabalhos futuros que envolvam a aplicação de tais técnicas e a elaboração de proposta de uma nova técnica que possa vir a contribuir com o desenvolvimento de *software*.

Bibliografia

CARVALHO, Ariadne Maria Brito Rizzoni; CHIOSSI, Telma Cecília dos Santos. **Introdução à Engenharia de *Software***. Campinas: Editora da Unicamp, 2001.

SOMMERVILLE, Ian. **Engenharia de *Software***. Tradução André Maurício de Andrade Ribeiro. São Paulo, Pearson Addison Wesley, 2003.

PRESSMAN, Roger. **Engenharia de *Software***. 5. ed. Rio de Janeiro: MacGraw-Hill,1995.

PETERS, James F.; PEDRYCZ, Witold. **Engenharia de Software – Teoria e Prática**. Tradução Ana Patrícia Garcia. Rio de Janeiro: Campus, 2001.