



**GOVERNO DO
ESTADO DO CEARÁ**

*Secretaria da Ciência, Tecnologia
e Educação Superior*



**UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ**

Engenharia de Software



Núcleo de Educação a Distância
Universidade Estadual Vale do Acaraú

Engenharia de Software

Vídeo

Engenharia de Requisitos: <http://youtu.be/UZfpUdYLsao>

Definição

Requisitos: São capacidades que um usuário necessita para resolver um problema ou atingir um objetivo (Thayer & Dorfman, 2000).

Exemplo: O sistema deve enviar um e-mail para o novo usuário assim que ele se cadastrar.

Um grande desafio enfrentado por um engenheiro de software é compreender os requisitos de um problema, isso se deve a dois fatores: o cliente não sabe o que é necessário para o sistema a ser desenvolvido, ou então, mesmo conhecendo as suas necessidades e estas se modificam no decorrer do projeto.

Curiosidade

Estudos mostram que os maiores problemas no desenvolvimento de software são relacionados com requisitos. (Standish Group, 1995)

Definição

Engenharia de requisitos: É a ciência e disciplina preocupada com a análise e documentação dos requisitos, incluindo análise das necessidades e análise e especificação dos requisitos, fornecendo mecanismos apropriados para facilitar as diversas atividades relacionadas. (Thayer & Dorfman, 2000.)

A engenharia de requisitos ajuda os engenheiros de software a compreender melhor o problema que eles vão trabalhar para resolver. Ela inclui o conjunto de tarefas que levam a um entendimento de qual será o impacto do software sobre o negócio, do que o cliente quer e de como os usuários finais vão interagir com o software.

Tarefas

O processo da engenharia de requisitos é composto de sete tarefas: concepção, levantamento, elaboração, negociação, especificação, validação e gestão de requisitos.

Concepção

A concepção se baseia na primeira reunião do engenheiro de software com o cliente. A intenção é estabelecer um entendimento básico do problema, o pessoal que quer uma solução, a natureza da solução desejada e a efetividade da comunicação e colaboração preliminares entre cliente e desenvolvedor.

Levantamento

O objetivo é ajudar o cliente a definir o que é necessário para a resolução do problema em questão, ou seja, os requisitos. Aparentemente uma tarefa simples, porém existem três problemas segundo Christel e Kang [CRI92] que tornam o levantamento uma tarefa difícil:

Problemas de escopo: O limite do sistema é mal definido ou o cliente/usuário especifica detalhes técnicos desnecessários que podem confundir, em vez de esclarecer, os objetivos globais do sistema.

Problemas de entendimento: Os clientes/usuários não estão completamente certos do que é necessário, têm pouca compreensão das capacidades e limitações de seu ambiente computacional, não têm pleno entendimento do domínio do problema, têm dificuldade do domínio do problema, têm dificuldade de informar as necessidades ao engenheiro de sistemas, omitem informações que acreditam ser “óbvias”, especificam requisitos conflitantes com as necessidades de outros clientes/usuários ou especificam requisitos que são ambíguos ou impossíveis de testar.

Problemas de volatilidade: Os requisitos mudam ao longo do tempo.

Elaboração

Definir o sistema em nível de implementação com base nas informações obtidas com o cliente. A elaboração é guiada pela criação e refinamento de cenários do usuário que descrevem como o usuário final (e outros atores) vai interagir com o sistema. Classes assim como seus respectivos métodos, atributos e relacionamentos

serão definidos.

O resultado final da elaboração é um modelo de análise que define o domínio do problema informacional, funcional e comportamental.

Negociação

A negociação se trata dos requisitos solicitados por seus clientes/usuários. É necessário estabelecer um consenso dos requisitos do sistema de forma que tanto o cliente quanto o engenheiro de software saiam ganhando.

Especificação

A especificação varia seu significado conforme pessoa e a complexidade do projeto, uma especificação pode ser um documento por escrito, um modelo gráfico, um modelo matemático formal, uma coleção de cenários de uso, um protótipo ou qualquer combinação desses elementos. A especificação descreve a função e o desenvolvimento do sistema e as restrições que governarão o seu desenvolvimento.

Validação

O objetivo da validação é corrigir ambiguidades, inconsistências, omissões e erros que venham a ocorrer na especificação dos requisitos. Essa tarefa é feita pelos engenheiros de software, clientes, usuários e demais interessados.

Gestão de Requisitos

A gestão de requisitos se baseia em um conjunto de atividades que ajudam a equipe de projetos a identificar, controlar e rastrear requisitos e modificações de requisitos em qualquer época, à medida que o projeto prossegue. A gestão de requisitos começa com a identificação. A cada requisito é atribuído um modo identificador. Uma vez identificados os requisitos, tabelas de rastreamento são desenvolvidas (Tabela 1). Cada tabela de rastreamento relaciona os requisitos identificados a um ou mais aspectos do sistema ou de seu ambiente. Exemplos de tabelas de rastreamento:

Tabela de rastreamento de características: Mostra como os requisitos se relacionam a características importantes do sistema/produto observáveis pelo cliente.

Tabela de rastreamento de fontes: Identifica a fonte de cada requisito.

Tabela de rastreamento de dependência: Indica como os requisitos estão relacionados uns aos outros.

Tabela de rastreamento de subsistemas: Caracteriza os requisitos pelo(s) subsistema(s) que eles governam.

Tabela de rastreamento de interface: Mostra como os requisitos se relacionam com as interfaces internas e externas do sistema.

Requisito	Aspecto do sistema				
	A1	A2	A3	A4	A5
R1		x	x		
R2	x				
R3		x			x
R4				x	

Tabela 1 - Tabela de Rastreamento Genérica

Início do Processo de Engenharia de Requisitos

Em um ambiente ideal, clientes e engenheiros de software trabalhariam juntos na mesma equipe. Sendo assim, a engenharia de requisitos seria apenas baseada em diálogos construtivos para o projeto. Mas na maioria dos casos esse ambiente não existe.

O cliente pode estar em outra localidade, ter apenas uma vaga ideia do que é necessário, ter opiniões conflitantes sobre o sistema a ser construído, ter conhecimento técnico limitado e tempo limitado para interagir com o engenheiro de requisitos.

Nas seções seguintes discutiremos os passos necessários para iniciar a engenharia de requisitos.

Identificação dos Interessados

Segundo Sommerville e Sawyer [SOM97] um interessado é “quem quer que se beneficie de modo direto ou indireto do sistema que está sendo desenvolvido”. Desde

já podemos identificar como interessado o próprio cliente. Demais interessados serão descobertos à medida que for feito o levantamento de requisitos, cabe ao engenheiro de software listar todos os interessados.

Reconhecimento de Diversos Pontos de Vista

Diferentes interessados terão diferentes pontos de vista, conforme as informações são coletadas os possíveis requisitos podem ser inconsistentes ou conflitar uns com os outros. O engenheiro de requisitos deve categorizar todas essas informações, afim de que seja feito um conjunto de requisitos consistente para compor o sistema.

Trabalho em busca da colaboração

O trabalho do engenheiro de requisitos é identificar áreas de concordância e áreas de conflito ou inconsistência entre os requisitos dos interessados e atingir um consenso, uma das maneiras é o uso de “Pontos de Prioridade”.

Pontos de Prioridade

A todos os interessados é fornecida certa quantidade de pontos de prioridade que podem ser “gastos” com qualquer número de requisitos. Uma lista de requisitos é apresentada e cada interessado indica a importância relativa de cada um (de acordo com o seu ponto de vista) gastando um ou mais pontos de prioridade com ele. Os pontos gastos não podem ser reusados. Quando os pontos de prioridade de um interessado se esgotam, nenhuma outra ação sobre requisitos pode ser tomada por essa pessoa. O total de pontos gastos em cada requisito por todos os interessados fornece uma indicação da importância global de cada requisito.

Interessado	Pontos de Prioridade	Requisitos			
		R 01	R 02	R 03	R 04
I 01	10	8	2	0	0
I 02	10	5	0	5	0
I 03	10	5	3	2	0
I 04	10	3	3	3	1
Total		21	8	10	1

Tabela 2 - Exemplo do uso de Pontos de Prioridade

Formulação das Primeiras Questões

O primeiro conjunto de questões livres de contexto focaliza o cliente e outros interessados, os objetivos globais e os benefícios.

As questões abaixo têm como intuito identificar interessados no software, os seus benefícios e possíveis alternativas para o desenvolvimento do software.

Quem está por trás da solicitação deste trabalho?

Quem vai usar a solução?

Qual será o benefício econômico de uma solução bem-sucedida?

Há outra fonte para a solução que você necessita?

Questões que permitem a equipe de software compreender melhor o problema e o cliente formular uma solução:

Como você caracteriza “boas” saídas, que seriam geradas por uma solução bem-sucedida?

Que problemas(s) essa solução enfrentaria?

Você pode me mostrar (ou descrever) o ambiente de negócios no qual a solução será usada?

Tópicos ou restrições especiais de desempenho afetarão o modo pelo qual a solução é abordada?

O conjunto final de questões focaliza a efetividade da própria atividade de comunicação. Gause e Weinberg[GAU98] as chamam de “metaquestões” e propõem a seguinte lista(resumida):

Você é a pessoa certa para responder a essas questões? Suas respostas são “oficiais”?

Minhas questões são relevantes ao problema que você tem?

Estou formulando muitas questões?

Alguém mais pode fornecer informações adicionais?

Devo perguntar-lhe mais alguma coisa?

Levantamento de Requisitos

Levantamento de requisitos é uma função importante na hora de desenvolver um sistema. Entender o que o cliente deseja ou o que o cliente acredita que precisa e as regras de negócios. Tudo isso faz parte desse levantamento durante um processo de desenvolvimento, que faz parte da Engenharia de Software.

Coleta Colaborativa de Requisitos

Levantamento de requisitos é uma ação coletiva, em que todos os envolvidos

no projeto sejam eles engenheiros de software, clientes e outras pessoas interessadas formem um conjunto para identificar o problema, definir e propor elementos de uma solução e analisar diferentes abordagens.

Diretrizes Básicas para uma Coleta Colaborativa de Requisitos:

- As reuniões devem ser conduzidas ou assistidas pela equipe de desenvolvedores e clientes juntamente com outros interessados. Nas reuniões são estabelecidas regras para preparação e participação dos integrantes. Para que os mesmo possam fazer um levantamento de pontos importantes a respeito do desenvolvimento do sistema.
- Será escolhido um ‘facilitador’, que pode ser qualquer um participante da reunião, que controlará a reunião.
- Um mecanismo de definição é usado, pode ser rascunho, papel auto-adesivo, salas de conversa, fóruns virtuais.

Tudo isso com um único objetivo de identificar o problema analisando diferentes pontos de vista e apresentar as melhores soluções.

Implantação da Função de Qualidade

A IFQ – Implantação da Função de Qualidade é uma técnica que transforma as necessidades dos clientes em requisitos técnicos de software. A IFQ se concentra em trazer a maior satisfação para o cliente.

A IFQ cobre todo o processo de engenharia de software, por isso ele é usado para um levantamento de requisitos, pois muitos de seus conceitos se aplicam a isso. As reuniões com os clientes serão usadas para identificar o valor de cada função necessária ao sistema, os objetos de dados e eventos que o sistema deverá produzir. Para isso, a IFQ identifica três tipos de requisitos:

- Requisitos Normais: Algo que o cliente possa ver que o projeto está de acordo com as metas e os objetivos estabelecidos. Por exemplo: Tipos de mostradores gráficos, funções específicas do sistema e níveis de desempenho.
- Requisitos Esperados: São requisitos implícitos no sistema ou produto fundamentais para a satisfação do cliente, apesar do cliente não cita-lo explicitamente. Ex.: Facilidade de interação homem/máquina e de instalação do software.
- Requisitos Excitantes: São os requisitos que superam as expectativas que o cliente tinha sobre o produto ou o sistema. Por exemplo, um software de processamento de texto é solicitado com as características padrão. O produto entregue dispõe de várias facilidades de layout de página que são muito agradáveis e inesperadas.

Cenários de Usuários

São um conjunto de cenários que serão construídos pelos engenheiros, clientes e usuários a fim de determinar uma linha de uso para o sistema a ser construído. Tudo isso será feito, pois ao decorrer do projeto, em que os desenvolvedores estão coletando os requisitos, fica mais difícil avançar para as funções mais técnicas de engenharia de software. Pois os mesmos terão dificuldade como as funções e características serão usadas por diferentes classes de usuários. Portanto esses cenários são construídos, frequentemente chamados de ‘Casos de Uso’, que irá fornecer detalhadamente a descrição do sistema.

Produtos de Trabalho do Levantamento

São produtos que serão produzidos em decorrência da coleta de requisitos. Muitos desses produtos podem variar pelo tamanho do projeto do sistema ou produto a ser desenvolvido. Mas na maioria dos projetos, os produtos de trabalho incluem:

- Uma declaração da necessidade e da viabilidade;
- Uma afirmação limitada do escopo do sistema ou do produto;
- Uma lista de clientes, usuários e outros interessados que participam do levantamento de requisitos;
- Uma descrição do ambiente técnico do sistema;
- Uma lista de requisitos e as restrições de domínios que se aplicam a cada um deles;
- Um conjunto de cenários de uso que fornecem informações sobre o uso do sistema ou do produto sob diferentes condições de operação;
- Quaisquer protótipos desenvolvidos para definir melhor os requisitos.

Cada um desses produtos será revisado pela equipe que participou do levantamento de requisitos.

Desenvolvimento de Casos de Uso

“Um caso de uso descreve o comportamento do sistema sob várias condições em que o sistema responde a solicitação de um de seus interessados”

Alistair Cockburn

Em outras palavras, um caso de uso conta uma história detalhada do usuário final

interagindo com sistema em suas diversas funções.

Os casos de usos possuem diversas características, por exemplo, definir o conjunto de atores que estarão envolvidos na história. Atores representam papéis que pessoas desempenham quando interagem com o sistema. Identificar quem são os atores principais e secundários. Atores primários interagem diretamente com o software, esperando dele um resultado satisfatório e benéfico. Atores secundários dão suporte ao sistema, de modo que os atores principais possam fazer seus trabalhos. Então ficam aqui alguns exemplos de perguntas essenciais para a criação de casos de uso:

- Quem é(são) o(s) ator(es) principal(is) e o(s) ator(es) secundários?
- Quais são as metas dos atores?
- Que pré-condições devem existir antes da história começar?
- Que tarefas ou funções principais são desempenhadas pelo ator?
- Que exceções deveriam ser consideradas quando a história é descrita?
- Que informações do sistema o ator vai adquirir, produzir ou modificar?
- Que informações o ator deseja do sistema?
- O ator deseja ser informado sobre modificações inesperadas?

Construção do Modelo de Análise

O objetivo do modelo de análise é fornecer uma descrição dos domínios informacional, funcional e comportamental necessários a um sistema baseado em computador. O modelo representa o atual instante na percepção geral do projeto e se mantém dinâmico para refletir mudanças nas necessidades dos interessados no projeto ou no entendimento dos engenheiros de software.

À medida que o modelo evolui, certos elementos se tornam relativamente estáveis, fornecendo uma base sólida para as tarefas posteriores de projeto. No entanto, outros elementos do modelo podem ser mais voláteis, indicando que o cliente ainda não entende totalmente os requisitos do sistema.

Elementos do Modelo de Análise

Dentre as muitas maneiras de se representar os requisitos de um sistema, alguns defendem a aplicação exclusiva de uma (casos de uso, por exemplo) em detrimento das outras. Outros acreditam no emprego simultâneo de vários modos de representação, pois essa abordagem tem maior probabilidade de descobrir omissões, inconsistências e ambiguidades.

O método de modelagem de análise define os elementos específicos do modelo, mas existem elementos genéricos comuns à maioria dos modelos de análise. São eles:

- Elementos baseados em cenário. Utilização de uma abordagem baseada em cenário para descrever o ponto de vista do usuário. Por exemplo, casos de uso básicos e seus respectivos diagramas evoluem para outros mais elaborados baseados em gabaritos.
- Elementos baseados em classe. Representam um conjunto de “objetos” com características e comportamento essencialmente iguais. Por exemplo, um diagrama de classe para descrever um tipo de “objeto” com seus respectivos atributos e operações sobre estes atributos.
- Elementos comportamentais. São elementos do modelo de análise que descrevem comportamento. Por exemplo, o diagrama de estados, que representa os estados e eventos que causam modificação do estado sistema.
- Elementos orientados a fluxo. Descreve todo o processo de transformação da informação de entrada na sua resposta apropriada através de um modelo de fluxo.

Padrões de Análise

São repetições dentro de um mesmo projeto ou entre projetos (geralmente de um mesmo domínio de aplicação) que permitem a reutilização (reuso) de um mesmo modelo para representá-las.

Podemos citar dois benefícios do uso de Padrões de Análise:

- Aceleram o desenvolvimento de modelos de análise abstratos que captam os principais requisitos do problema, fornecendo uma descrição das vantagens e limitações.
- Facilitam a transformação do modelo de análise em um modelo de projeto, sugerindo padrões de projeto e soluções e soluções confiáveis para problemas comuns.

Padrões de análise são integrados ao modelo de análise referenciando o nome do padrão e são representados segundo um gabarito padrão com a seguinte forma:

- Nome do Padrão: um descritor que capta a essência do padrão.
- Intenção: descreve o que o padrão realiza ou representa.
- Motivação: um cenário que ilustra como o padrão pode ser utilizado.
- Forças e contexto: descreve tópicos externos que podem afetar o padrão e/ou serem resolvidos com a aplicação do mesmo.
- Solução: descreve como o padrão resolve o problema com ênfase nos tópicos estrutural e comportamental.
- Consequências: implicações e/ou compromissos decorrentes da aplicação do padrão.
- Projeto: discute como se chegar ao padrão por meio de padrões de projeto. Como Adapter ou Singleton, por exemplo.
- Usos conhecidos: exemplos de usos em sistemas reais.
- Padrões relacionados: padrões que se relacionam com o padrão em questão, seja no uso, na estrutura ou porque derivam do mesmo.

Negociação de requisitos

Desenvolvimento de um plano de projeto que satisfaça as necessidades do cliente, ao mesmo tempo em que reflete as restrições (como prazo, pessoal orçamento etc.) do mundo real propostas para a equipe de software.

As negociações buscam satisfazer da melhor maneira possível todas as partes envolvidas. Para tanto, é importante seguir três passos:

1. Identificar os interessados-chave do sistema ou subsistema.
2. Determinar as “condições de ganho” dos interessados.
3. Negociação dessas condições, reconciliando-as em um conjunto de condições “ganha-ganha” para todos os envolvidos.

A conclusão bem-sucedida desses passos deve atingir um resultado “ganha-ganha”, que será o critério-chave para a sequência das atividades de engenharia de software.

Validação dos requisitos

Cada elemento do modelo de análise é examinado quanto a consistência, omissões e ambiguidade. Os requisitos do modelo são então priorizados (pelo cliente, preferencialmente) e agrupados em pacotes para serem implementados e entregues ao cliente. Numa revisão do modelo de análise, as seguintes questões devem ser tratadas:

- Cada requisito está consistente com o objetivo global do sistema/produto?
- Algum requisito conflita com outro?
- O modelo de requisitos reflete adequadamente a informação, a função e o comportamento do sistema a ser construído?

Essas e muitas outras questões devem ser formuladas e respondidas para garantir que o modelo de requisitos reflita adequadamente as necessidades do cliente e forneça uma base sólida para o projeto.