

SOFT

DISCIPLINA: Engenharia de Software

AULA NÚMERO: 7

DATA: ____/____/____

PROFESSOR: Andrey

APRESENTAÇÃO

Nesta aula serão apresentados e discutidos os conceitos de Técnicas de extração de requisitos de um sistema de software.

DESENVOLVIMENTO

Técnicas para extração de requisitos (Carvalho, 2001)

- Técnicas informais: baseadas em comunicação estruturada, interação com o usuário, análise de documentos, questionários etc. Exemplos: *Joint Application Design* (JAD), *Brainstorming*, *PIECES* e entrevistas.
- Técnicas formais: pressupõem a construção de um modelo conceitual do problema ou de um protótipo do software. Exemplos: modelo funcional, modelo de dados, modelo de objetos.

Entrevistas

É uma técnica de extração de requisitos que é realizada através de uma série de encontros com os usuários. Nestes encontros os usuários respondem perguntas sobre o seu trabalho, o ambiente no qual atuam, suas necessidades, entre outras. É uma técnica que requer do desenvolvedor algumas habilidades sociais como habilidade de ouvir e conhecimento de técnicas de entrevistas. Um entrevistador competente pode ajudar a entender e explorar os requisitos do software de forma a amenizar os problemas vistos anteriormente.

A entrevista consta de 4 fases: Identificação dos entrevistados, preparação da entrevista, condução da entrevista e finalização da entrevista.

A identificação dos entrevistados normalmente começa com o financiador do projeto (cliente) ou com os usuários do sistema a ser desenvolvido. Não é necessário que todos os entrevistados sejam identificados antes de começarem as entrevistas. Outros entrevistados podem ser identificados durante o processo.

A preparação para a entrevista envolve agendar a entrevista, deixar claros os objetivos e a duração da entrevista, preparar material para a entrevista e preparar uma lista de questões que servirá de base para a entrevista.

A condução da entrevista é feita da seguinte forma: é feita uma introdução revisando os objetivos e formato da entrevista. A seguir a entrevista segue tópico por tópico, explorando cada tópico com perguntas que abordem diferentes direções e diferentes níveis de abstração. A entrevista deve ser guiada de forma a evitar que o entrevistado se concentre em muitos detalhes ou uma única solução para um problema. Deve-se ter o cuidado para sumarizar sempre o que já foi tratado de forma a verificar a correta compreensão e evitar erros de comunicação como erros de observação, de foco, de interpretação, conflitos de opiniões, ambiguidades ou fatos que não sejam verdadeiros. A entrevista pode terminar quando as questões tiverem sido respondidas, por tempo ou cansaço do entrevistado.

Ao final da entrevista é feita uma sumarização e consolidação da informação recebida descrevendo os tópicos que foram abordados e os que ficaram pendentes. Pode-se dar a oportunidade para o

entrevistado revisar e corrigir um resumo escrito da entrevista.

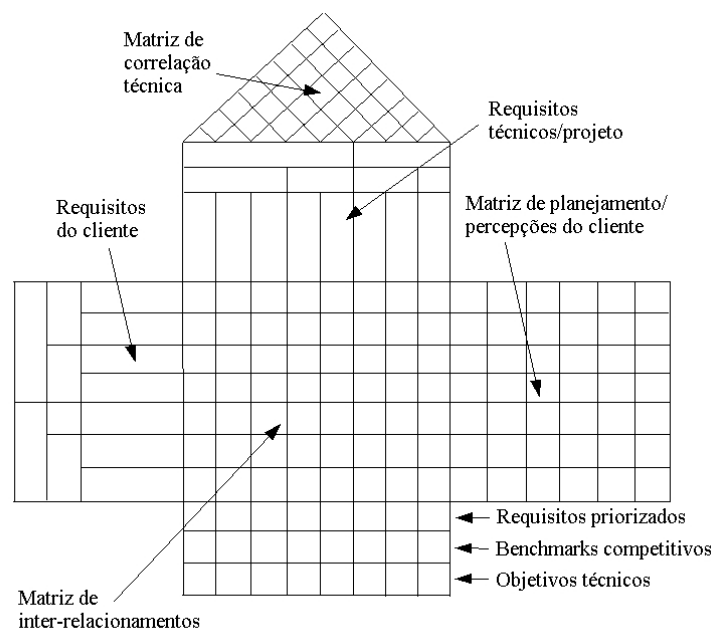
Desdobramento da Função de Qualidade (QFD)

O método do desdobramento da função de qualidade (QFD) foi desenvolvido na empresa Mitsubishi no Japão por Y. Akao e S. Mizuno. Com a aplicação do QFD e HOQ nos anos entre 1977 e 1984, a Toyota reduziu os custos de desenvolvimento de produto em 61%, o ciclo de desenvolvimento em um terço e virtualmente eliminou os problemas de garantia relacionado com ferrugem.

A Casa da Qualidade (HOQ) é uma matriz que traduz o conjunto dos requisitos do cliente, pesquisas de mercado e informações de comparações técnicas em um conjunto de características de engenharia organizadas de acordo com sua prioridade que serão usadas no desenvolvimento de um novo produto.

A Casa da Qualidade funciona da seguinte maneira:

- As necessidades do cliente são organizadas na forma de uma lista na região dos requisitos do cliente.
- Estes requisitos do cliente são quantificados em termos de prioridade e percepção de desempenho em relação a outros produtos na região da matriz de planejamento.
- Os requisitos técnicos do produto são listados e organizados na região de requisitos técnicos. A região da matriz de correlação técnica é usada para representar como os requisitos técnicos interferem uns com os outros.
- Em seguida as relações entre os requisitos do cliente e os requisitos técnicos são quantificadas em termos de sua importância para o projeto na matriz de inter-relacionamentos. As conclusões são sumarizadas na parte de prioridades, comparações e alvos técnicos.



Estrutura da casa da qualidade (HOQ)

Uma técnica para aplicar QFD para o desenvolvimento de *software* é a SQFD. O objetivo da *Software Quality Function Deployment* (SQFD) é melhorar o desenvolvimento de *software* aplicando técnicas de melhoria de qualidade durante a especificação de requisitos. Para isto, as necessidades dos clientes são confrontadas com as restrições próprias do projeto de forma a concentrar os melhores esforços nos aspectos com maior importância.

PIECES

Conjunto de categorias de perguntas que ajudam na extração de requisitos

- **P** erformance: identifica as tarefas que o produto irá executar e então o tempo de resposta para cada tipo de tarefa (*throughput*).
- **I** nformações e dados: identifica a qualidade e quantidade de informações as quais os clientes terão acesso.
- **E** conomia: questões relacionadas à demanda.
- **C** ontrole: questões relacionadas ao acesso às informações.
- **E** ficiência: espécie de relação do tipo custo-benefício visando minimizar desperdícios.
- **S** erviços: os usuários respondem perguntas sobre que tipo de serviço eles precisam que o produto realize.

BRAINSTORMING

Técnica não estruturada para geração de idéias que consiste em duas fases:

- geração de idéias: idéias são apresentadas sem discussão
 - Não permitir críticas ou debates;
 - deixar a imaginação fluir
 - gerar o número máximo de ideias possível
 - modificar e combinar ideias
- consolidação: idéias são discutidas, revisadas e organizadas
 - “podar” as ideias que não são relevantes para o projeto
 - Agrupar as ideias
 - Definir características
 - Priorizar ideias

Joint Application Design (JAD) - Projeto Aplicado Conjunto

Conjunto de técnicas para promover cooperação, entendimento e trabalho em equipe entre usuários e desenvolvedores a fim de se obter uma melhor extração de requisitos.

- Ocorre por meio de reuniões planejadas, conduzidas de forma neutra, embaladas por técnicas de dinâmica de grupo e documentadas em tempo real. Ao final das sessões, terão sido cristalizados os modelos de eventos e de dados.
- As sessões JAD são pilotadas por um condutor neutro especializado em técnicas de tomada de decisão em grupo, especificação de sistemas e gerência de encontros.
- O time de desenvolvimento de sistemas comparece com seus analistas, para quem os usuários finais explicam as suas necessidades, processos e sua realidade de negócios.
- O linguajar deve ser simples e destituído de jargões a fim de facilitar o processo de comunicação entre os clientes e os desenvolvedores (PMBOK, 2000).

São princípios do JAD (abordagem Carvalho, 2001):

- dinâmica de grupo
- uso de técnicas visuais
- manutenção do processo organizado e racional
- utilização de documentação padrão

O JAD é dividido em duas etapas: Planejamento (extração e especificação de requisitos) e Projeto. Cada etapa consiste em 3 fases:

- **Adaptação.** Fase que consiste na preparação para a sessão, com as atividades de organizar as equipes, adaptar a técnica ao produto e preparar o material. Participam: líder da sessão,

engenheiro de requisitos, executor, representante dos usuários, representante de produtos de software e o especialista.

- Sessão. Fase onde os requisitos são elicitados. Composta das seguintes atividades: Conduzir orientações, definir requisitos de alto nível, delimitar o escopo do sistema, documentar questões e considerações e concluir a fase de sessão.
- Finalização. Fase onde as anotações, transparências, e outros documentos são transformados em documentos de especificação. Composta das seguintes etapas: Completar o documento, Revisar o documento e Obter a aprovação do executor.

Roteiro para Um documento de Especificação dos Requisitos

1. Introdução

1.1. Objetivos

Propósito da especificação

- Usuários previstos para especificação

1.2. Escopo

- Nome do produto a ser construído
- O que o software produto fará (e se necessário o que não fará)
- Objetivos do produto e benefícios relevantes

1.3. Referência a documentos citados

- Lista de todos os documentos referenciados na especificação, indicando o título, data de publicação, organização que publicou.

1.4. Visão geral da especificação

- Organização da especificação
- Conteúdo das demais seções

2. Descrição geral

2.1. Projeto com relação a outros sistemas

Neste item deve ser informado se o software:

- É independente e totalmente autocontido, ou.
- Componente de um sistema mais amplo; nesse caso, deve-se incluir:

a. as funções de cada componente do software

b. as principais interfaces externas do software

c. o hardware e periféricos utilizados.

A utilização de um diagrama mostrando os principais componentes do sistema, as interligações entre os componentes e as interfaces entre os componentes e as interfaces externas ajuda ao entendimento do software que está sendo especificado.

2.2. Funções do software

- Sumário das funções a serem realizadas pelo software

2.3. Características dos usuários

- Características gerais dos eventuais usuários do sistema, as quais possam afetar os requisitos específicos, tais como: nível de escolaridade, experiências e formação técnicas, entre outras

2.4. Restrições gerais

- Descrição de outros itens que impõem restrições/limitações ao projeto, tais como: padrões utilizados, limitações de hardware, interfaces com outras aplicações, etc.

2.5. Considerações e dependências

- Indicação dos fatores cuja alteração afetará os requisitos especificados, como por exemplo, se a especificação pressupõe o uso de um sistema operacional específico, a sua não disponibilidade implicará em alterações ao documento de especificação.

3. Requisitos funcionais

3.1. Modelo de casos de uso

- Os atores do sistema
- Os casos de uso identificados para o sistema

3.1.1. Diagrama de casos de uso

Diagrama com os casos de uso do sistema

Em seguida, para cada um dos casos de uso identificados, fornecer:

3.1.1. Descrição do caso de uso

- Breve descrição do que faz o caso de uso

3.1.2. Descrição dos atores

- Nome do ator
- Breve descrição

3.1.3. Fluxos

- Especificação dos fluxos de execução do caso de uso contendo, entre outras informações:
 1. Descrição do fluxo de eventos básico.
 2. Descrição dos subfluxos
 3. Descrição dos fluxos alternativos, pontos de extensão.
 4. Cenários

4. Requisitos de interfaces externas

4.1. Interfaces com o usuário

- formatos das telas
- "layout" e conteúdo das páginas de relatórios e menus apresentados no vídeo
- tempo máximo que o usuário tem para responder a uma saída do sistema

4.2. Interfaces de hardware

Características de cada interface entre o software e os componentes de hardware, podendo incluir informações tais como: equipamentos a serem utilizados, protocolos utilizados, etc.

4.3. Interfaces de software

Este parágrafo deve especificar o uso de outros produtos de software, tais como: sistema de gerenciamento de banco de dados, pacote de rotinas matemáticas, etc.

Para cada produto, informar:

- nome
- mnemônico
- número de especificação

- versão

Para cada interface:

- Descrever o relacionamento entre o software desenvolvido e o software ao qual ele interfaceia
- Definir a interface em termos do conteúdo e formato das mensagens

5. Requisitos de desempenho

5.1. Requisitos numéricos estáticos

- número de terminais suportados
- número de usuários simultâneos suportados
- número de arquivos e registros a serem manuseados
- tamanho de tabelas e arquivos

5.2. Requisitos numéricos dinâmicos

Incluem: tempo de resposta do sistema a um estímulo externo, número de transações, tarefas e quantidade de dados a serem processados em um certo período de tempo, em condições normais ou de alta carga do sistema

6. Atributos

6.1. Segurança

Este parágrafo especifica os fatores que protegerão o software contra o acesso, uso, modificações, destruição acidental ou intencional. Requisitos específicos nessa área podem incluir a necessidade de:

- Utilizar técnicas de criptografia
- Manter um "log" ou conjunto de dados de histórico
- Restringir a comunicação entre partes do programa
- Incluir código de verificação("checksum") para valores críticos

6.2. Confiabilidade/disponibilidade

Este parágrafo especifica os fatores que garantem um nível definido de confiabilidade/disponibilidade para o sistema, o que inclui mecanismos de:

- Detecção de erros: testes de validação dos dados de entrada, paridade,
- Recuperação de erros: pontos de verificação ("checkpoint"), reconfiguração do sistema.

6.3. Facilidade de manutenção

- Requisitos que asseguram a existência de condições para que software possua manutenção eficaz

6.4. Transferência/conversão para outros sistemas

- Mecanismos e procedimento necessários para transportar o sistema de um ambiente para outro

7. Restrições de projeto

- Concordância a padrões ou regulamentos já existentes
- Limitações de hardware: limites de memória principal ou secundária, características da configuração do hardware

8. Outros requisitos

8.1. Banco de dados

Especificar requisitos para o banco de dados a ser desenvolvido como parte de produto, o que pode incluir:

- tipos de informações
- relações entre elementos de dados entre elementos de dados, registros e arquivos

Se um banco de dados vai ser utilizado, ele deve ser incluído nos Requisitos de interfaces externas, mas seu uso deve ser descrito aqui.

8.2. Operações

- Funções normais e especiais requeridas pelos usuários; um exemplo são operações de "backup" e recuperação.

8.3. Adaptação ao ambiente operacional

Características relacionadas com o local ou missão, que devem ser modificadas para adaptar o software a uma determinada instalação.

Observações:

A fim de facilitar o rastreamento dos requisitos e a verificação de sua implementação, sugere-se que a cada requisito seja associado um código. Tal código pode ser constituído do seguinte modo:

1. Associar a cada tipo de requisito uma identificação de duas letras; assim:

- requisitos funcionais - RF
- requisitos de desempenho - RD
- requisitos de interfaces - RI
- restrições de projeto - RP
- atributos - AT
- outros requisitos - OR

2. numerar os requisitos pertencentes a cada tipo; assim:

- RF01 identifica o requisito funcional de número 01
- RI12 identifica o requisito de interface de número 12

ATIVIDADE

1) Faça uma especificação de requisitos para o sistema descrito abaixo:

Sistema de Matrícula em cursos de Universidade

O processo de designação de professores para cursos e a matrícula de estudantes é uma experiência frustrante e que consome muito tempo.

Depois que os professores da Universidade decidem em quais cursos eles vão lecionar no semestre, a Secretaria alimenta essa informação no sistema em computador. Um relatório *batch* é impresso para os professores indicando em quais cursos eles lecionarão. Um catálogo de cursos é impresso e distribuído para os estudantes.

Atualmente, os alunos preenchem formulários de matrícula que indicam suas escolhas de cursos e retornam os formulários para a Secretaria. A carga típica de um estudante é de quatro cursos. Os funcionários da Secretaria alimentam os dados dos formulários dos estudantes no sistema de computador no *mainframe*. Uma vez que o currículo para o semestre foi alimentado no sistema, um *job batch* é executado durante a noite para inscrever os estudantes nos cursos. Na maioria das vezes os estudantes obtêm sua primeira escolha; entretanto, naqueles casos em que há um conflito, a Secretaria conversa com cada estudante para obter escolhas adicionais. Quando todos os estudantes tiverem sido inscritos com sucesso nos cursos, um relatório dos currículos dos estudantes é encaminhado para eles

UNIVERSIDADE FEDERAL DO PARANÁ
ESPECIALIZAÇÃO EM INFORMÁTICA

para verificação. A maioria dos registros dos estudantes são processados durante uma semana, mas alguns casos excepcionais demoram até duas semanas para serem solucionados.

Quando o período inicial de matrícula é concluído, os professores recebem uma lista de estudantes para cada curso que devem lecionar.

O novo sistema vai permitir, no início de cada semestre, que os estudantes possam solicitar um catálogo de cursos contendo uma lista dos cursos oferecidos no semestre. Informações sobre cada curso, tais como professor, departamento e pré-requisitos estarão incluídas para ajudar os estudantes a tomarem suas decisões.

O novo sistema vai permitir aos estudantes selecionarem quatro dos cursos oferecidos para o semestre. Além disso, cada estudante indicará duas escolhas alternativas para caso um curso oferecido seja cancelado ou não tenha vagas suficientes. Nenhum curso poderá ter mais de dez ou menos do que três alunos. Um curso com menos de três alunos será cancelado. Uma vez concluído o processo de matrícula de um estudante, o sistema de matrícula envia informação para o sistema de faturamento de forma que o aluno possa receber as faturas para o semestre.

Professores devem poder acessar o sistema *online* para indicar quais cursos eles lecionarão e para ver que estudantes se inscreveram em suas ofertas de curso.

Para cada semestre, há um período de tempo no qual os estudantes podem mudar sua grade horária. Os estudantes devem poder acessar o sistema durante este período para adicionar ou retirar cursos.

2) Faça a especificação de requisitos para o sistema, tema do seu trabalho da disciplina.

BIBLIOGRAFIA BÁSICA

PRESSMAN, R. S.. Engenharia de Software. Makron Books. 1995

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.. UML guia do usuário. Editora Campus. 2000.

BEZERRA, E.. Princípios de Análise e Projeto de Sistemas com UML. Editora Campus. 2003.

CARVALHO, A. M. B. R.; CHIOSSI, T. C. S.. Introdução à engenharia de software. Editora da Unicamp. 2001.