

REQUISITOS DE SOFTWARE

ARIADNE CARVALHO (ÚLTIMA ALT. 10/8/2010)

Requisito

- condição necessária para a obtenção de certo objetivo ou para o preenchimento de certo fim.

Requisitos para um sistema de software

- descrição das funções e restrições que o produto a ser desenvolvido deve possuir.

Engenharia de Requisitos

- processo de descobrir, analisar, documentar e verificar essas funções e restrições.

Tipos de Requisitos

1. **Do usuário:** declarações, em língua natural e/ou diagramas, sobre as funções que o sistema deve fornecer e restrições sob as quais deve operar. São requisitos abstratos de alto nível.
2. **Do sistema:** funções e restrições do sistema, de uma forma mais detalhada. Normalmente classificados como funcionais e não-funcionais.

Requisitos funcionais

- Diretamente ligados à funcionalidade do software, como o sistema deve reagir à entradas específicas, como deve se comportar em determinadas situações.
- Em alguns casos podem declarar o que o sistema **não** deve fazer.
- Dependem do tipo de sistema a ser desenvolvido e dos usuários.

- **Exemplo:** Sistema de biblioteca de universidade → permite pedir livros e documentos a outras universidades.
- (a) buscar todo o conjunto inicial no banco de dados ou selecionar um subconjunto;
- (b) fornecer telas apropriadas para ler documentos no repositório de documentos;
- (c) alocar um único identificador a cada pedido.

Requisitos funcionais (cont.)

- (a) descritos em diferentes níveis de detalhes
(telas apropriadas = diferentes formatos);
- (b) documento completo e consistente, mas na prática é quase impossível atingir essa meta.
- (c) à medida que os problemas são descobertos, o documento de especificação deve ser corrigido.

Requisitos não-funcionais

- Não dizem respeito diretamente às funções específicas do sistema.
- Podem estar relacionados às propriedades do sistema como *dependability*, tempo de resposta, restrições sobre o processo, padrões, etc.

- Exemplos:
 - (a) dependendo do resultado do teste, somente o supervisor pode efetuar a entrada do resultado do teste de um paciente.
 - (b) o sistema deve emitir um recibo para o cliente até oito segundos após a transação.
 - (c) um sistema de aviação deve atender ao requisito de *dependability*.
 - (d) um sistema de tempo real deve atender ao requisito de desempenho; do contrário as funções de controle não operarão corretamente.
 - (e) tipos de ferramentas CASE e descrição do processo a ser seguido.

Requisitos não-funcionais (cont.)

1. Requisitos do produto: comportamento do produto - desempenho, memória, *reliability* (taxa aceitável de falha), portabilidade e facilidade de uso;
2. Requisitos organizacionais: políticas e procedimentos nas organizações do cliente e do desenvolvedor - padrões de processo, requisitos de implementação (linguagem ou método de projeto) e requisitos de entrega (do produto e documentos associados);
3. Requisitos externos: fatores externos ao sistema e ao processo de desenvolvimento - interoperabilidade (com outros sistemas), requisitos legais, requisitos éticos.

Objetivo do sistema versus requisitos verificáveis

- Objetivo: O sistema deve ser fácil de utilizar por controladores experientes e deve ser organizado de modo que os erros dos usuários sejam minimizados;
- Requisito verificável: Controladores experientes devem ser capazes de utilizar todas as funções do sistema depois de duas horas de treinamento. O número médio de erros cometidos por usuários experientes não deve exceder a dois por dia.

Requisitos de domínio

- Tem origem no domínio de aplicação e refletem características desse domínio;
- Podem ser novos requisitos funcionais, podem restringir requisitos funcionais existentes, ou ainda estabelecer como realizar cálculos específicos.

Exemplo para o sistema de biblioteca:

1. Deve haver uma interface padrão com o usuário para todos os bancos de dados, que terá como base o padrão X.
2. Em razão das restrições referentes a direitos autorais, alguns documentos devem ser imediatamente excluídos após serem fornecidos.
3. Alguns documentos serão impressos localmente no servidor do sistema para serem encaminhados ao usuário ou direcionados para uma impressora de rede.

Requisitos do usuário

- Devem descrever os requisitos funcionais e não-funcionais de modo compreensível pelos usuários sem conhecimento técnico detalhado.
- Devem especificar o comportamento externo do sistema, evitando características de projeto.
- Podem ser escritos em língua natural, formulários e diagramas intuitivos simples.

Problemas com uso de língua natural

1. Falta de clareza: ambigüidade e falta de precisão, dando origem a um documento de difícil leitura;
2. Confusão: os requisitos funcionais e não-funcionais, os objetivos do sistema e as informações sobre o projeto podem não estar claramente definidos;
3. Fusão de requisitos: vários requisitos diferentes podem ser expressos juntos, como um único requisito.

Documento de Especificação de Requisitos

- Declaração oficial do que é exigido dos desenvolvedores do sistema.
- Inclui os requisitos do usuário para um sistema e uma especificação detalhada dos requisitos do sistema.
- O documento tem um número diversificado de usuários:

Doc. de Requisitos de Software (cont.)

- (a) Clientes do sistema: especificam e verificam se os requisitos atendem as suas necessidades; também especificam mudanças;
- (b) Gerentes: usam o documento para planejar o processo de desenvolvimento;
- (c) Engenheiros de software: compreender que sistema deverá ser desenvolvido;
- (d) Engenheiros de teste: desenvolver testes de validação do sistema;
- (e) Engenheiros de manutenção: compreender o sistema e as relações entre suas partes.

Documento de Especificação de Requisitos de Software

1. **Introdução:** Descreve os objetivos do sistema, suas funções e explica como ele deve operar com outros sistemas. Descreve também como o sistema se ajusta aos objetivos estratégicos da organização que está encomendando o software.
2. **Glossário:** Define os termos técnicos utilizados no documento. Não se deve fazer suposições sobre a experiência ou conhecimento do leitor.

Documento de Especificação de Requisitos (cont.)

3. **Definição dos requisitos do usuário:** Descreve os serviços fornecidos para o usuário e os requisitos não funcionais do sistema. Pode-se utilizar língua natural, diagramas ou outras notações que sejam compreendidas pelos clientes.

4. **Especificação dos requisitos do sistema:**

Descreve os requisitos funcionais e não funcionais com mais detalhes. Podem ser definidas interfaces, isto é, como o software interage com as pessoas, com o hardware do sistema, com outros sistemas e com outros produtos. As restrições impostas pela aplicação, tais como padrões, linguagem de implementação, ambientes operacionais e limites de recursos também são descritas.

5. **Evolução do sistema:** Descreve as suposições fundamentais nas quais o sistema se baseia e as mudanças previstas devido à evolução do hardware, mudanças nas necessidades do usuário, etc.
6. **Análise de risco:** Define os pontos de risco e as ações a serem contempladas para evitar ou minimizar impacto dos riscos.
7. **Anexo:** Descreve todos os recursos e técnicas utilizados para a o levantamento de requisitos, assim como as questões feitas e o nome das pessoas envolvidas.

Processos da Engenharia de Requisitos

1. Estudo de viabilidade
2. Levantamento e análise dos requisitos
3. Validação dos requisitos
4. Gerenciamento dos requisitos

Estudo de viabilidade

- O estudo de viabilidade permite que se decida se vale a pena desenvolver o sistema proposto.
- O sistema contribui para os objetivos da organização?
- Pode ser implementado com a tecnologia atual e dentro do orçamento?
- Pode ser integrado com outros sistemas em operação?

Estudo de viabilidade (cont.)

- O que aconteceria se o sistema não fosse implementado?
- Quais os problemas com os processos atuais?
- Como o sistema proposto irá ajudar?
- Pode haver troca de informações entre outros sistemas e o sistema proposto?

Levantamento e análise dos requisitos

- Desenvolvedores trabalham com o cliente e usuários finais para descobrir mais informações sobre o domínio da aplicação, serviços, desempenho, restrições de hardware, etc.
- Envolve diferentes tipos de pessoas.
- *Stakeholder*: qualquer pessoa com alguma influência, direta ou indireta, sobre os requisitos.
- Exemplo: usuários finais, todo pessoal afetado pelo sistema; desenvolvedores, mantenedores de sistemas relacionados, gerente de negócios, especialistas no domínio, etc.

Problemas com o levantamento

1. Os usuários muitas vezes não sabem o que querem, a não ser em termos muito gerais: podem achar difícil articular o que desejam do sistema, fazer pedidos não realistas.
2. Os usuários expressam os requisitos em seus próprios termos e com conhecimento implícito de sua área de atuação. Engenheiros de requisitos devem entender esses requisitos.

Problemas com o levantamento (cont.)

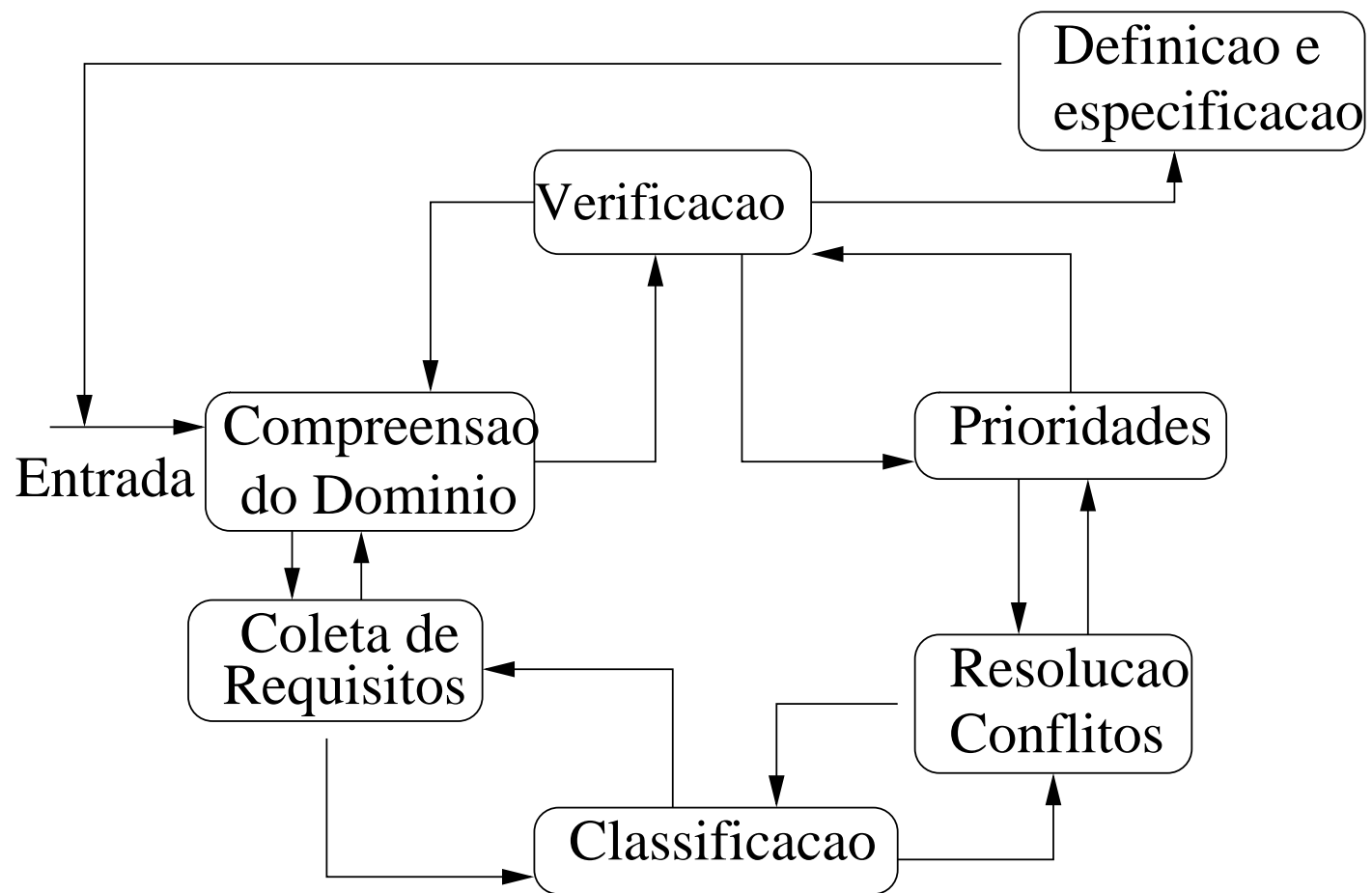
3. Diferentes usuários tem em mente diferentes requisitos e podem expressá-los de maneira distinta. Os engenheiros de requisitos devem descobrir todas as fontes possíveis e encontrar pontos comuns e conflitos.
4. O ambiente econômico e de negócios é dinâmico e se modifica durante o processo de análise → a importância dos requisitos pode mudar, novos requisitos podem surgir.

O processo de levantamento de requisitos

1. *Compreensão do domínio*: documentos, livros, sistemas, pessoas. Exemplo: sistema de biblioteca → entender como funcionam as bibliotecas.
2. *Coleta e análise de requisitos*: descoberta, revelação e entendimento dos requisitos, através de interação entre clientes, usuário(s) e desenvolvedores envolvendo:
 - a descoberta, classificação e organização dos requisitos;
 - a determinação de suas prioridades;
 - resolução de inconsistências e conflitos; e
 - descoberta de omissões.

O processo de levantamento de requisitos (cont.)

3. *Especificação dos requisitos*: armazenamento dos requisitos em uma ou mais formas, incluindo língua natural, linguagem semiformal ou formal, representações simbólicas ou gráficas (casos de uso, por exemplo);
4. *Validação dos requisitos*: verificação dos requisitos, visando determinar se estão completos e condizentes com as necessidades e desejos do usuário.



Descrição de um sistema hospitalar

GOSTARIA QUE FOSSE CONSTRUÍDO UM SISTEMA PARA MONITORAR A TEMPERATURA E A PRESSÃO DE PACIENTES DA UTI, QUE DEVERÃO FICAR LIGADOS *on-line* À REDE DE COMPUTADORES DO HOSPITAL, QUE É FORMADA POR UM COMPUTADOR PRINCIPAL E VÁRIOS TERMINAIS QUE MONITORAM OS PACIENTES. SE A TEMPERATURA OU PRESSÃO DO PACIENTE LIDA PELO TERMINAL SE TORNAREM CRÍTICAS, O COMPUTADOR PRINCIPAL DEVERÁ MOSTRAR UMA TELA DE ALERTA COM UM HISTÓRICO DAS MEDIDAS REALIZADAS PARA O PACIENTE.

Descrição de um sistema hospitalar (cont.)

UM AVISO SONORO DEVE SER ATIVADO NESSE CASO. A VERIFICAÇÃO DA PRESSÃO É FEITA COMPARANDO-SE A PRESSÃO DO PACIENTE COM UM VALOR PADRÃO DE PRESSÃO (MÁXIMO E MÍNIMO) A SER DIGITADO PELO RESPONSÁVEL E VERIFICANDO SE A PRESSÃO MEDIDA ESTÁ DENTRO DOS PARÂMETROS CONSIDERADOS NORMAIS PARA O PACIENTE (VALORES PRÓXIMOS AO MÁXIMO E MÍNIMO SÃO PERMITIDOS). TEMOS VÁRIOS SISTEMAS *on-line* NO COMPUTADOR E TODOS DEVEM RODAR AO MESMO TEMPO.

Funções:

- monitorar temperatura e pressão; e
- apresentar uma tela de alerta com o histórico de medidas.

Restrições:

- o sistema deve ser *on-line*;
- deve rodar ao mesmo tempo que outros → controle de concorrência; e
- o aviso de temperatura e pressão críticas deve ser sonoro.

Ambigüidades no sistema hospitalar

Se a temperatura ou pressão do paciente lida pelo terminal se tornarem críticas, o computador principal deverá mostrar uma tela de alerta com um histórico das medidas realizadas para o paciente. Um aviso sonoro deve ser ativado nesse caso.

- Duas interpretações:
 1. tela de alerta: onde será mostrada? No computador principal e/ou no terminal do paciente?
 2. aviso sonoro: o computador principal ativará um aviso sonoro? E o terminal do paciente?

Omissões do sistema hospitalar

A verificação da pressão é feita comparando-se a pressão do paciente com um valor padrão de pressão (máximo e mínimo) a ser digitado pelo responsável e verificando se a pressão medida está dentro dos parâmetros considerados normais para o paciente (valores próximos ao máximo e mínimo são permitidos).

1. o que significa valores próximos?
2. quais são os valores possíveis para máximo e mínimo?
3. o que acontece com valores fora de um parâmetro normal?

- Mudanças nos requisitos acontecem na maioria dos sistemas complexos.
- Embora muitas delas sejam devidas a mudanças das necessidades dos usuários, outras advêm da interpretação incorreta dos requisitos do produto a ser desenvolvido.
- REQUISITOS INCOMPLETOS, INCORRETOS OU MAL ENTENDIDOS SÃO AS CAUSAS MAIS FREQUENTES DA BAIXA QUALIDADE, ULTRAPASSAGEM DOS CUSTOS PREVISTOS E ATRASO NA ENTREGA DO PRODUTO DE SOFTWARE.

BRAINSTORMING

- Técnica básica para geração de idéias.
- Uma ou várias reuniões que permitem que as pessoas sugiram e explorem idéias sem que sejam criticadas ou julgadas.
- Existe um *líder* cujo papel é fazer com que a sessão comece, sem restringí-la.

- Especialmente útil no começo do processo de extração de requisitos pois:
 - ausência de crítica e julgamento ajuda a eliminar algumas das dificuldades inerentes ao processo.
 - evita a tendência a limitar o problema muito cedo.
 - fornece uma interação social mais confortável do que algumas técnicas de grupo mais estruturadas.
 - pode ser aprendida, com muito pouco investimento.
- Desvantagem: por ser um processo relativamente não estruturado, pode não produzir a mesma qualidade ou nível de detalhe de outros processos.

1. Geração de idéias

- Participantes fornecem idéias, sem discussão sobre o mérito delas.
- Útil na geração de várias visões do problema e na sua formulação de diferentes maneiras.
- Atividades dessa fase:
 - identificação dos participantes (normalmente usuários e desenvolvedores);
 - designação do líder;
 - agendamento da sessão com todos os participantes;
e
 - preparação da sala.

Geração de idéias (cont.)

- Saída: depende das idéias geradas (pessoas com conhecimento e especialidades apropriados).
- Líder abre a sessão falando sobre o problema de um modo geral, e os participantes podem gerar novas idéias para expressar o problema.
- Continua enquanto novas idéias estiverem sendo geradas.

Geração de idéias (cont.)

- Quatro regras:
 1. é terminantemente proibido criticar as idéias;
 2. idéias não convencionais ou estranhas são encorajadas;
 3. o número de idéias geradas deve ser bem grande; e
 4. os participantes devem ser encorajados a combinar ou enriquecer as idéias de outros (idéias visíveis).

Geração de idéias (cont.)

- A fase de geração pode terminar de duas maneiras:
 1. se o líder acreditar que não estão sendo geradas idéias suficientes.
 2. se tiverem sido geradas e registradas idéias suficientes.

2. Consolidação das idéias

- Idéias são discutidas, revisadas, organizadas e avaliadas.
- Algumas idéias são rephraseadas.
- Quando duas ou mais idéias são consideradas iguais, são combinadas e reescritas para capturar a sua essência.
- Os participantes podem concordar em que algumas das idéias são muito esquisitas e descartá-las.

Consolidação das idéias (cont.)

- Idéias remanescentes são discutidas e classificadas em ordem de prioridade.
- Freqüentemente é necessário identificar:
 - requisitos absolutamente essenciais;
 - aqueles que são bons, mas não essenciais; e
 - aqueles que seriam apropriados para uma versão subsequente do software.
- O líder ou outra pessoa designada produz um registro das idéias remanescentes, juntamente com suas prioridades ou outros comentários relevantes.

ENTREVISTAS

- Série de encontros com os usuários que explicam:
 - o seu trabalho;
 - o ambiente no qual atuam;
 - as suas necessidades etc.
- Técnica estruturada, que pode ser aprendida e na qual os desenvolvedores podem ganhar proficiência.
- Requer o desenvolvimento de algumas habilidades sociais gerais:
 - habilidade de ouvir; e
 - conhecimento de uma variedade de táticas de entrevista.

Fases da Entrevista

1. Planejamento da entrevista;
2. Condução da entrevista; e
3. Finalização.

Planejamento da entrevista

- Ler material disponível
- Estabelecer objetivo da entrevista:
 - frequência dos serviços do novo sistema
 - previsibilidade dos serviços
 - atualidade dos dados
- Decidir quem será entrevistado
 - incluir uma pessoa-chave de cada nível afetado
 - pedir ajuda na empresa para a escolha de pessoas

Planejamento da entrevista (cont.)

- Preparar os entrevistados
 - avisar a data e duração
 - comunicar o assunto
- Preparar lista de questões
 - direcionadas para o objetivo da entrevista
 - informações obtidas → novas questões

Tipos de questões

abertas-dirigidas: → “Explique como esse relatório é produzido”

Vantagem → descobre-se detalhes e vocabulário

Desvantagem → perde-se a objetividade e gasta-se tempo

fechadas: “Quantos relatórios desse tipo são gerados por mes?”

Vantagem: facilidade na compilação dos resultados

Desvantagem: falta de detalhes e monotonia

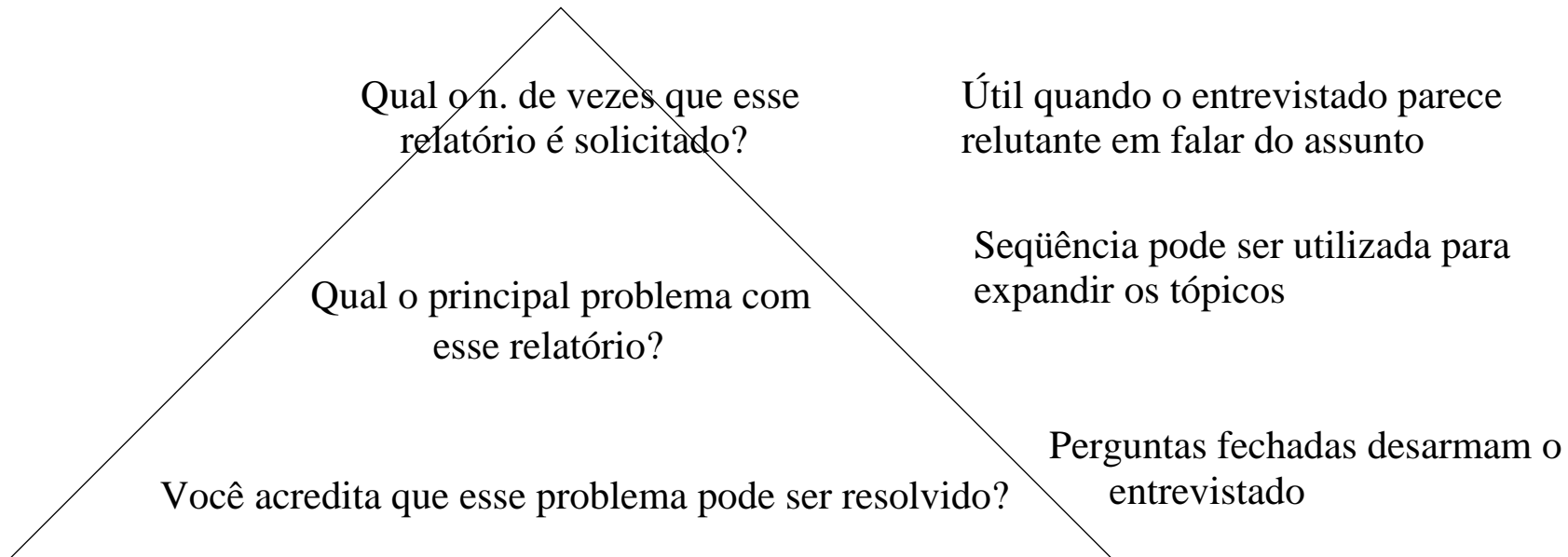
seqüência: dá continuidade a uma questão. “Por que? Dê um exemplo.”

Estrutura da entrevista

- **Pirâmide**

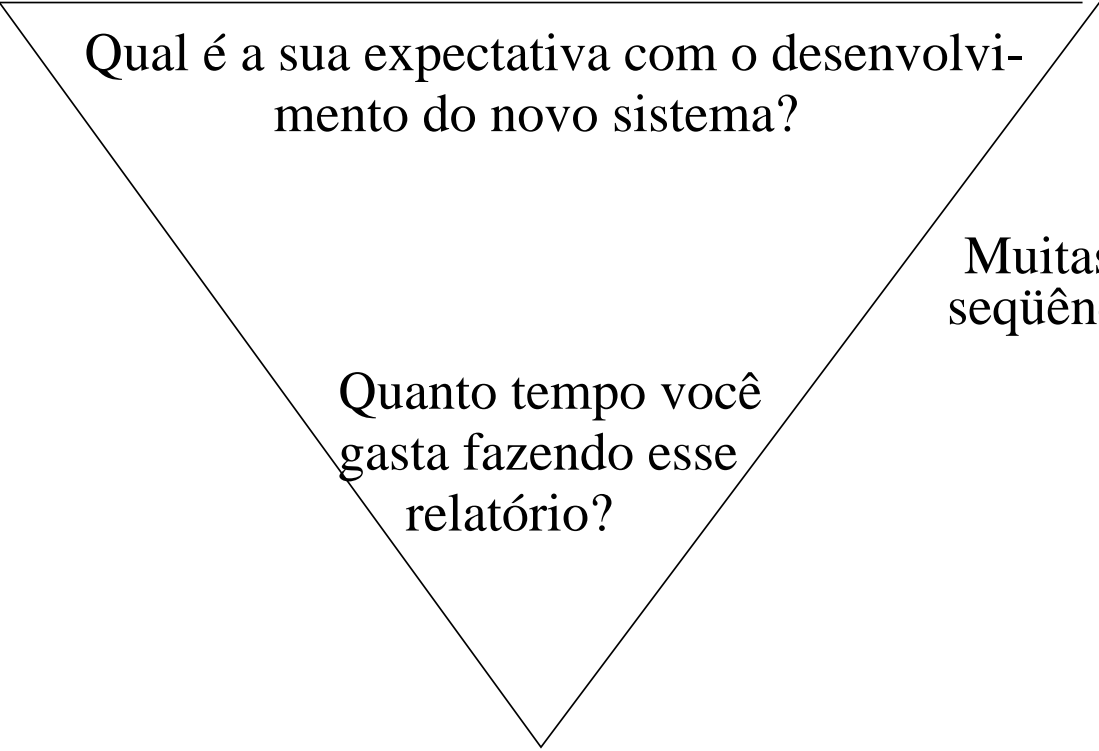
começa com questões fechadas → obtém respostas diretas

expande os tópicos com questões abertas dirigidas



- **Funil**

começa obtendo detalhes → questões abertas dirigidas
dá continuidade obtendo respostas diretas → questões fechadas

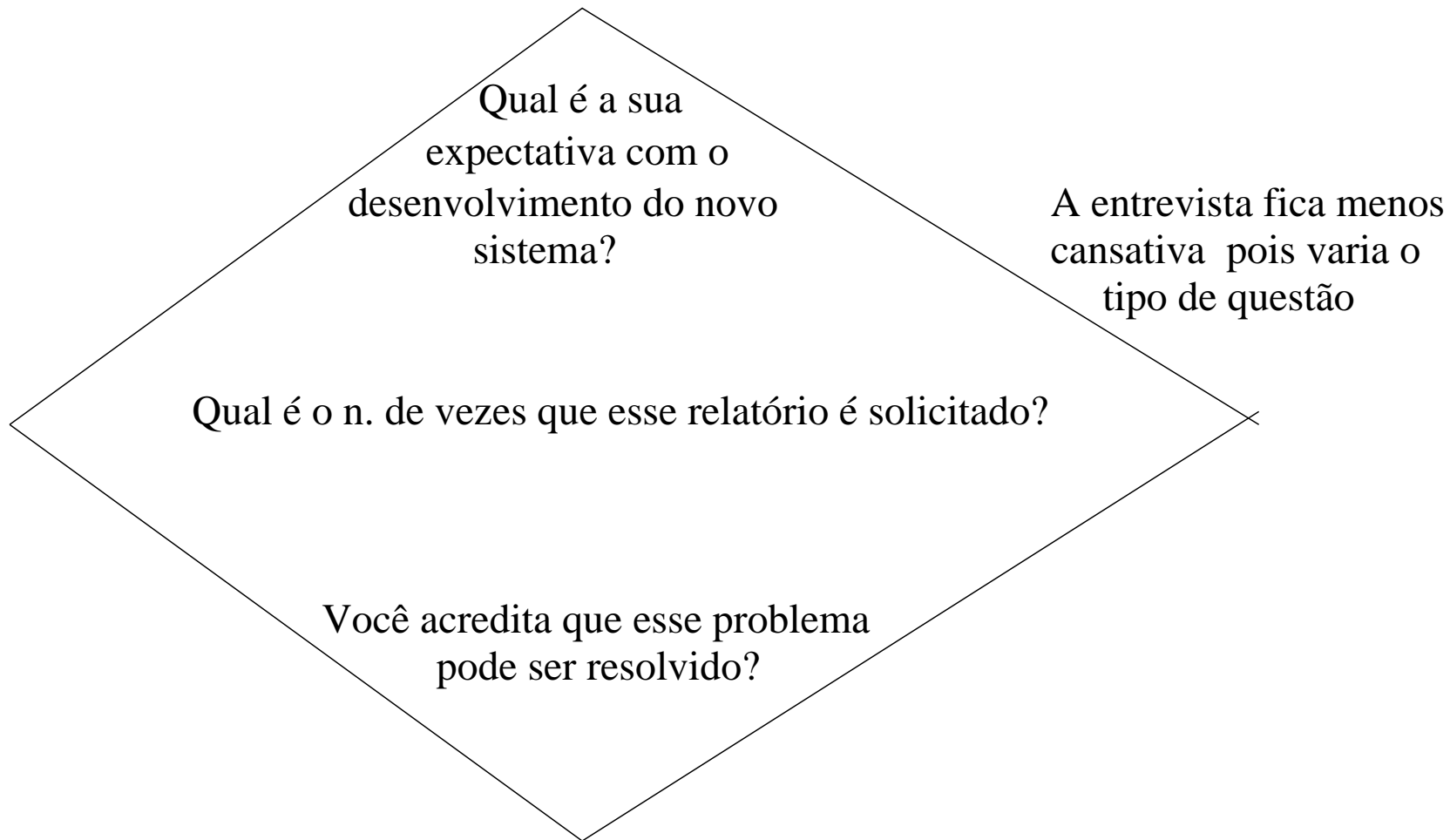


Qual é a sua expectativa com o desenvolvimento do novo sistema?

Quanto tempo você
gasta fazendo esse
relatório?

Muitas questões fechadas e
seqüências tornam-se necessárias

- **Diamante** → combina as duas estruturas anteriores



Finalização da entrevista

- Quando todas as questões tiverem sido feitas e respondidas;
- Quando o tempo alocado tiver se esgotado; ou
- Quando o entrevistador sentir que o entrevistado está exausto.

Finalização da entrevista (cont.)

- Reservar cinco ou dez minutos para sumarizar e consolidar a informação recebida (principais tópicos explorados e aqueles que necessitam de informação adicional).
- Explicar as próximas ações a ser tomadas, incluindo a oportunidade para o entrevistado revisar e corrigir um resumo escrito da entrevista.
- Agradecer o entrevistado pelo tempo e esforço dedicados.

Atividades após a entrevista

- Enviar ao entrevistado um agradecimento por escrito.
- Produção de um resumo escrito → reconhecer ou reordenar os tópicos discutidos e consolidar a informação obtida:
 - descobrir ambigüidades; e
 - informação conflitante ou ausente.
- Informações estatísticas ou baseadas em fatos relatados de memória → confirmar com fontes confiáveis.
- Revisar procedimentos utilizados para preparar e conduzir a entrevista → melhorar o processo.

Habilidades e estratégias para comunicação oral

- A primeira resposta para a pergunta pode não estar necessariamente completa e correta.
- Pode ser expressa numa linguagem desconhecida para o entrevistador (resumir, rephrasear e mostrar as implicações do que o entrevistador está ouvindo).
- A sumarização é útil durante a entrevista toda e não só no final (confirma o entendimento, generalizações úteis e abstrações de alto nível).

Habilidades e estratégias (cont.)

- Questões específicas: não induzir respostas como “O relatório de vendas deveria ser produzido semanalmente?”.
- Perguntas com respostas do tipo “sim” ou “não” permitem que o entrevistado responda sem que precise de muito tempo para pensar.
- Uma única pergunta sobre um determinado tópico pode não produzir uma resposta completa ou significativa.
- Explorar os tópicos com questões que os abordem em diferentes níveis de abstração.

Erros mais comuns

- **Erros de observação:** pessoas diferentes se concentram em diferentes aspectos e podem “ver” coisas diferentes.
- **Erros de memória:** o entrevistado pode estar confiando demais na lembrança de informações específicas, e a memória humana pode falhar.
- **Erros de interpretação:** o entrevistador e o entrevistado podem estar interpretando palavras comuns de maneira diferente, tais como “pequena quantidade de dados” ou “caracteres especiais”.

Erros mais comuns (cont.)

- **Erros de foco:** o entrevistador pode estar pensando de maneira ampla, e o entrevistado pode estar pensando de maneira restrita (ou vice-versa), o que afeta o nível de abstração na discussão daquele tópico.
- **Ambigüidades:** há ambigüidades inerentes à maioria das formas de comunicação, especialmente a língua natural.

Erros mais comuns (cont.)

- **Conflitos:** entrevistador e entrevistado podem ter opiniões conflitantes sobre um determinado problema, e a tendência é registrar o ponto de vista do entrevistador.
- **Fatos que simplesmente não são verdadeiros:** o entrevistado pode dar informações que ele assume como fatos verdadeiros, mas que, na verdade, são só a sua opinião.

QUESTIONÁRIO

- Forma rápida de se obter dados de uma grande amostra de usuários
- Tipos de dados que podem ser coletados:
 - a utilização do sistema atual
 - problemas que os usuários enfrentam em seu trabalho
 - expectativas dos usuários em relação ao novo sistema

- É apropriado quando:
 - as pessoas envolvidas estão dispersas (exemplo: filiais)
 - o número de pessoas envolvidas é muito grande
 - deseja-se explorar várias opiniões
 - deseja-se conhecer melhor o sistema para organizar melhor as entrevistas

Questionário

- As questões devem ser claras → não é possível explicá-las
- As possíveis respostas devem ser antecipadas
- A aplicação e compilação dos resultados devem ser planejadas antecipadamente

Tipos de questões

- **Questões abertas-dirigidas:** ‘Por que você acha que os manuais do usuário para o sistema de contabilidade não funcionam?’
 - antecipar o tipo de resposta (enumerá-las)
 - deve ser possível interpretar corretamente as respostas
 - utilizadas quando não é possível listar todas as alternativas

- **Questões fechadas:** “Os dados sobre vendas são normalmente entregues com atraso?”
 - utilizada quando é possível listar todas as alternativas
 - as respostas devem ser mutuamente exclusivas

Abertas

Fechadas

Lenta



Velocidade de Término



Rápida

Alta



Natureza Exploratória



Baixa

Alta



Largura e Profundidade



Baixa

Fácil



Facilidade de Preparação



Difícil

Difícil



Facilidade de Análise



Fácil

Linguagem empregada nos questionários

- Usar a linguagem de quem vai responder o questionário sempre que possível, mantendo as perguntas simples, claras e curtas.
- Ser específico, mas não exageradamente.
- Fazer a pergunta certa para a pessoa certa.
- Ter certeza de que as questões estão tecnicamente corretas antes incluí-las no questionário.

Elaboração do Questionário

- Ordem em que as perguntas devem aparecer.
- Questões mais importantes devem vir primeiro.
- As questões de conteúdo semelhante e relacionado devem estar próximas.
- As associações prováveis devem ser antecipadas pelo elaborador do questionário.
- As questões que podem gerar controvérsias devem ser deixadas para depois.

Aplicação do Questionário

- Quem responderá o questionário? → depende dos objetivos.
 1. Todos respondem ao mesmo tempo no mesmo lugar.
 2. Entregues pessoalmente e depois recolhidos.
 3. Colocados a disposição e depois devolvidos.
 4. Enviados por correio eletrônico ou correio normal (prazo e instruções de retorno).
 5. Entregue pelo engenheiro de requisitos.

Uso de escalas no questionário

→ Atribuição de números ou outros símbolos

- **Escala Nominal:** usada para classificar atributos ou características. Exemplo: Que tipo de programa você mais usa?
 1. processador de textos
 2. planilha eletrônica
 3. gerenciador de banco de dados
 4. programas gráficos

- **Ordinal:** classifica atributos ou características em uma determinada ordem.

Exemplo: A pessoa de suporte na empresa é:

1. muito útil
2. moderadamente útil
3. inútil

- **Intervalo**

- o intervalo entre as alternativas de resposta é igual
- Exemplo: Dê uma nota de 1 a 5 para o atendimento do pessoal de manutenção (1 para ruim e 5 para excelente)

- **Proporção**

- alternativas em termos de proporção ou %
- o intervalo entre as alternativas é igual
- existe o valor zero que representa a ausência do atributo.
- Exemplo: Qual o tempo aproximado que você trabalha no computador diariamente.
a) 0% b) 25% c) 50% d) 75% e) 100%

PIECES

- Desenvolvedores inexperientes dificilmente sabem como começar.
- Que perguntas fazer para extrair os requisitos.
- Seis categorias de problemas que podem ajudar o analista a estruturar o processo:
 1. *desempenho* (ou *performance*);
 2. *informação e dados*;
 3. *economia*;
 4. *controle*;
 5. *eficiência*; e
 6. *serviços*.

- Pode ser adaptada para incluir questões iniciais ou básicas que sejam especialmente relevantes para o tipo de software.
- Ajuda a lidar com dificuldades de articulação dos problemas e comunicação.
- Mais proveitosa na análise de produtos já existentes (manuais ou automatizados).
- Pode ser adaptada para domínios de aplicação específicos.
- Com a experiência: um conjunto de questões detalhadas pode ser elaborado (produtos novos e produtos a ser melhorados).

1. Desempenho

- Medido de duas maneiras:
 1. pelo número de tarefas completadas em uma unidade de tempo (*throughput*), tal como o número de pedidos processados no dia; e
 2. pelo tempo de resposta, ou seja, a quantidade de tempo necessária para executar uma única tarefa.
- Perguntas que ajudem a identificar as tarefas e o tempo de resposta para cada tipo de tarefa.
- Quando o produto já existe: descobrir se os usuários experientes já sabem onde existem problemas de desempenho.

2. Informação e dados

- Produtos de software fornecem dados ou informações úteis para a tomada de decisão.
- O software deve fornecer acesso:
 - ao tipo certo de informação (nem de mais nem de menos);
 - no tempo certo; e
 - em forma utilizável.
- Se os usuários tendem a não utilizar o produto → sintoma de que informações erradas estão sendo fornecidas.

- Se eles o utilizam, mas expressam frustração → o sistema apresenta muita informação, ou o faz de uma forma diferente daquela que o usuário necessita.
- Exemplo:
 - (1) relatório diário que seria necessário somente mensalmente, ou mensal que seria necessário diariamente.
 - (2) o relatório pode conter informação relevante, mas é preciso consultar um relatório de cem páginas várias vezes ao dia (acesso *on-line*).

3. Economia

- Custo de usar um produto de software é sempre importante.
- Dois fatores de custo inter-relacionados:
 1. nível de serviço: medida do desempenho do sistema (*throughput*, tempo de resposta, ou ambos).
 2. capacidade de lidar com alta demanda: em alguns sistemas varia consideravelmente de minuto a minuto, ou de hora em hora.
- Usuários gostariam de ter um nível de serviço ou desempenho relativamente estáveis.

- Pode-se embutir no produto a capacidade de lidar com a alta demanda necessária nas horas de pico:
 - Processadores adicionais, unidades de disco ou conexões de rede, projeto de estruturas de dados internas para armazenar informações de tamanho ou complexidade não previsíveis de tempos em tempos.
- Pode ser caro, e, portanto, essas questões devem ser discutidas com os usuários.
- Um completo entendimento da carga esperada e do nível de serviço necessário ao produto ajudará os desenvolvedores a tomar decisões.

4. Controle

- Sistemas são normalmente projetados para ter desempenho e saídas previsíveis.
- Quando o sistema se desvia do desempenho esperado → algum controle deve ser ativado para tomar ações corretivas.
- Em sistemas de tempo real → o controle é exercido diretamente pelo software.
- Segurança → controle importante para alguns produtos (acesso restrito a certos usuários ou a certas horas do dia).

- Tipo de acesso restrito (somente leitura ou leitura e escrita).
- Auditoria → habilidade de ver, monitorar ou reconstruir o comportamento do sistema, durante ou depois da execução do processo.
- Questões de controle são importantes para não construir:
 - um sistema que fornece pouco controle (processo pode fugir de controle); ou
 - Controle em excesso (impedir que o trabalho seja executado).

5. Eficiência

- Não é sempre que a energia e os recursos aplicados a uma tarefa produzem trabalho útil.
- Algumas vezes há uma perda.
- Eficiência → medida dessa perda (relação entre os recursos que resultam em trabalho útil e o total dos recursos gastos).
- Eficiência versus economia:
 - para melhorar a economia do processo, a quantidade de recursos deve ser reduzida;
 - para melhorar a eficiência, a perda no uso desses recursos deve ser reduzida.

- Algumas ineficiências podem ser caracterizadas como redundâncias desnecessárias:
 - Coletar o mesmo dado mais de uma vez, armazená-lo em espaços múltiplos ou computar um determinado valor mais de uma vez, uso de algoritmos e estruturas de dados pobres.
 - Interface pobre pode ocasionar perda de tempo do usuário.

6. Serviços

- Produtos de software fornecem serviços aos usuários.
- Pode ser útil pensar em termos de serviços durante o processo de extração de requisitos.
- Usuários respondem perguntas sobre que tipos de serviços eles precisam que o produto realize e como esses serviços devem ser fornecidos.
- O produto pode também prestar serviços a outros produtos de software → que interfaces serão necessárias entre esses dois produtos.

Levantamento orientado a pontos de vista

- Há diferentes tipos de usuário final, com diferentes interesses.
- Exemplo: Sistema de caixa automático de um banco (ATM):
 1. *Clientes do banco*: recebem serviços do sistema;
 2. *Representantes de outros bancos*: acordos de reciprocidade que permitem utilizar ATMs uns dos outros;

3. *Gerentes de agências bancárias*: obtêm informações do sistema;
4. *Equipes de atendimento de balcão*: envolvidas nas operações diárias do sistema, reclamações de clientes etc;
5. *Administradores de bancos de dados*: responsáveis pela integração do sistema com o banco de dados do cliente do banco;

6. *Gerentes de segurança bancária*: que devem garantir que o sistema não apresente nenhuma falha de segurança;
7. *Departamento de marketing*: interessado em utilizar o sistema como instrumento de marketing do banco;
8. *Engenheiros de manutenção de hardware e software*: fazer a manutenção do hardware e do software.

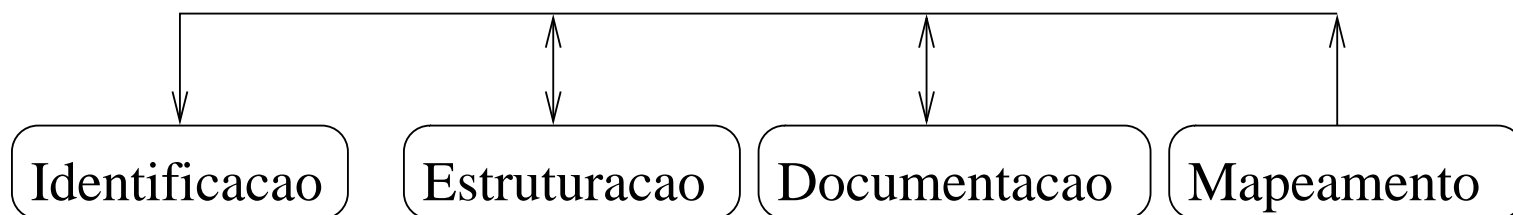
Pontos de vista - Vantagens

1. Como os pontos de vista são externos ao sistema, são uma maneira natural de estruturar o processo de levantamento de requisitos.
2. É relativamente fácil decidir se alguma coisa é um ponto de vista válido. Os pontos de vista devem interagir com o sistema de alguma maneira.
3. Os pontos de vista e os serviços são um meio útil de estruturar os requisitos não-funcionais. Cada serviço pode ter requisitos não-funcionais associados. Os pontos de vista permitem que o mesmo serviço tenha diferentes requisitos não-funcionais.

VORD - Definição de requisitos orientada a ponto de vista

1. Identificação dos pontos de vista: descobrir os pontos de vista que utilizam quais serviços específicos.
2. Estruturação dos pontos de vista: agrupar pontos de vista relacionados, segundo uma hierarquia. Serviços comuns localizados no nível mais alto e herdados por pontos de vista de nível inferior.

3. Documentação do ponto de vista: refinar a descrição dos pontos de vista e serviços identificados.
4. Mapeamento do sistema conforme pontos de vista (identificar objetos, utilizando informações de serviço encapsuladas nos pontos de vista).



- VORD: usa formulários-padrão para pontos de vista e serviços.
- Exemplo: ATM - sistema de software embutido, destinado a dirigir o hardware e se comunicar com a central de dados do banco.

- Aceita solicitações do cliente e fornece dinheiro, informações sobre conta, atualização de informações, etc.
- Clientes podem fazer retiradas, pagamentos, conferir saldos transferir dinheiro de uma conta para outra, pedir extrato, talão, etc.
- Máquinas de um banco podem permitir que clientes de outros bancos utilizem um subconjunto de seus recursos (retirada em dinheiro e consulta a saldo).

TEMPLATES VORD

Template de ponto de vista

Referencia: Nome do ponto de vista

Atributos: Informacoes sobre o ponto de vista

Eventos: Estimulos externos gerados pelo ponto de vista

Servicos: O que o sistema oferece

Subpontos de vista: Nomes dos pontos de vista associados

Template de servico

Referencia: Nome do servico

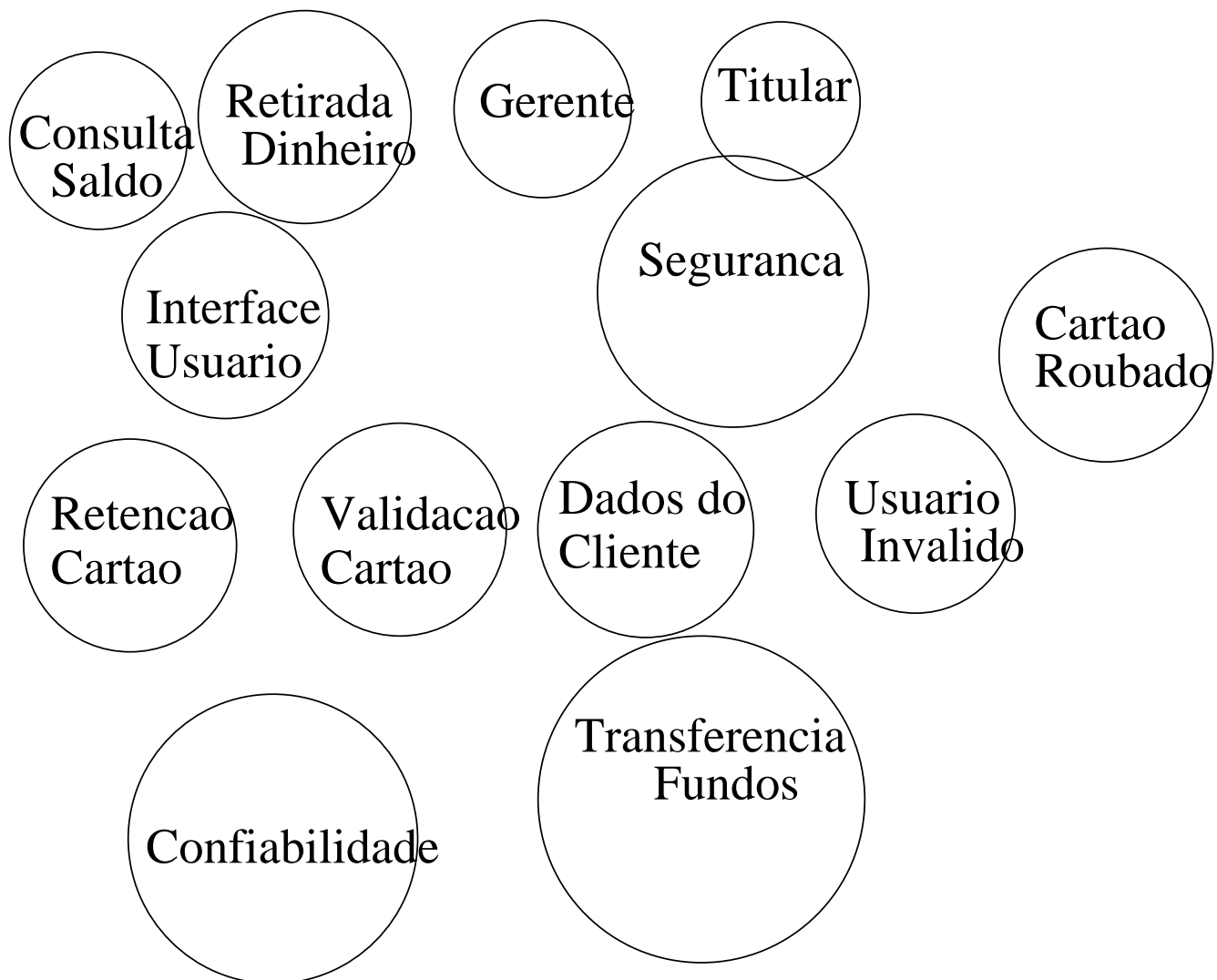
Razao: razao pela qual o servico e oferecido

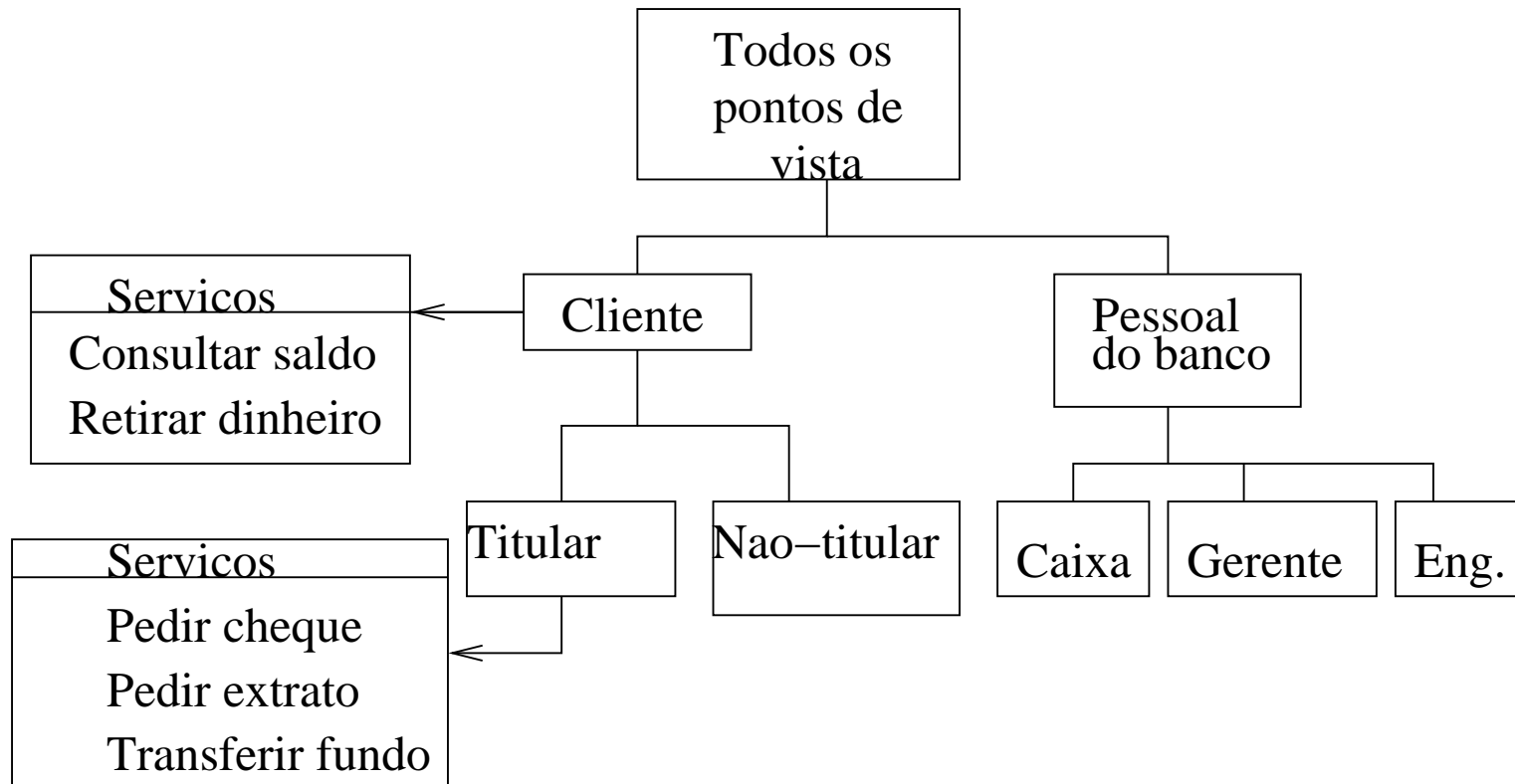
Especificacao: lista de especificacao de servicos

Pontos de vista: que recebem o servico

Requisitos nao funcionais: restricoes ao servico

Provedores: objetos que fornecem o servico





ALOCAÇÃO DE SERVIÇOS PARA OS PONTOS DE VISTA

- **Titular da conta:**

- Retirar dinheiro

- Consultar saldo

- Pedir extrato

- Transferir fundos

- **Nao-titular da conta:**

- Retirar dinheiro

- Consultar saldo

PONTO DE VISTA CLIENTE E RETIRADA DE DINHEIRO

Referencia: Cliente

Atributos: n. conta
PIN
início da transacao

Eventos: selecionar servico
cancelar transacao
encerrar transacao

Servicos: retirar dinheiro
consultar saldo

Subpontos : titular
de vista não titular

Referencia: retirar dinheiro

Razao: melhorar o servico

Especificacoes: pressionar botao
de retirada; em seguida informar
quantia solicitada; operacao con-
firmada se houver saldo

Ponto de vista: cliente

Req. n. funcional: entregar o
dinheiro um minuto apos
confirmada quantia

Provedor: preenchido posteri-
ormente

Validação dos requisitos

1. Verificação de validade: identificar funções adicionais ou diferentes.
2. Verificação de consistência: não devem existir requisitos conflitantes, restrições contraditórias ou diferentes para uma mesma função.
3. Verificação de completude: todas as funções e restrições exigidas pelo usuário.

Validação dos requisitos (cont.)

4. Verificação de realismo: com conhecimento da tecnologia existente, verificar se os requisitos realmente podem ser implementados (orçamento e prazos).
5. Facilidade de verificação: requisitos escritos de modo a serem verificados.

Técnicas de validação

Revisões de requisitos: formais e informais

- Revisão informal: envolve os desenvolvedores e tantos *stakeholders* quantos possível para discutir os requisitos.
- Revisão formal: a equipe de desenvolvimento deve:
 - “conduzir” o cliente, mostrando implicações de cada requisito.
 - revisores verificam cada um em termos de consistência, e os requisitos como um todo, em termos de completude.

Revisões (cont.)

- Também podem checar:
 - Facilidade de verificação: pode ser testado?
 - Facilidade de compreensão: pelos usuários ou compradores.
 - Facilidade de rastreamento: a origem do requisito é claramente definida? (pode ser preciso retornar a origem do requisito para avaliar o impacto de uma mudança)
 - Adaptabilidade: o requisito é adaptável? (isto é, modificável sem provocar efeitos em larga escala em outros requisitos)

Revisões (cont.)

- Conflitos, contradições, erros e omissões devem ser detectados e descartados durante a revisão e formalmente registrados.
- Os usuários, compradores e desenvolvedores devem negociar a solução para esses problemas.

Geração de casos de teste

- Requisitos devem ser testáveis.
- Quando os testes são criados como parte do processo de validação, podem revelar problemas.
- Se um teste é difícil ou impossível de ser projetado → requisitos de difícil implementação.

Gerenciamento de requisitos

- Requisitos estão sempre sendo modificados para sistemas complexos.
- Como o problema não pode ser inteiramente definido, os requisitos são necessariamente incompletos.
- Durante o processo de desenvolvimento, a compreensão dos desenvolvedores está em constante modificação, que se reflete nos requisitos.

Gerenciamento de requisitos (cont.)

- Com sistemas existentes: difícil prever que efeitos o sistema “atualizado” terá sobre a organização.
- Depois que os usuários finais se familiarizam com um sistema, novos requisitos surgem porque:

Gerenciamento de requisitos (cont.)

1. Comunidade de usuários diversificada, diferentes prioridades e requisitos, muitas vezes conflitantes ou contraditórios. Requisitos finais → conciliação entre eles.
2. Pessoas que pagam pelo sistema são diferentes das que usam. Restrições organizacionais e orçamentárias, conflitantes com os requisitos dos usuários.
3. A empresa e o ambiente técnico do sistema se modificam → refletido no sistema (novo hardware, novas interfaces com outros sistemas, prioridades da empresa mudam, novas legislações).

Gerência de requisitos (cont.)

- É o processo de compreender e controlar as mudanças nos requisitos do sistema.
- É realizado em conjunto com outros processos da engenharia de requisitos.
- O planejamento se inicia junto com o levantamento inicial de requisitos.
- O gerenciamento deve começar assim que uma versão inicial do documento de requisitos estiver disponível.

Requisitos permanentes e voláteis

1. **Requisitos permanentes:** relativamente estáveis, derivam da atividade principal da organização e se relacionam diretamente com o domínio.

Exemplo: Em um hospital, sempre haverá requisitos relativos aos pacientes, médicos, enfermeiras e aos tratamentos.

2. **Requisitos voláteis:** requisitos que provavelmente vão se modificar durante o desenvolvimento ou depois que o sistema estiver em operação.

Exemplo: Requisitos resultantes de políticas governamentais sobre assistência médica.

Classificação dos requisitos voláteis

1. **Requisitos mutáveis:** que se modificam devido a mudanças no ambiente no qual a organização opera. Exemplo: Em sistemas hospitalares, o financiamento do tratamento de pacientes pode se modificar e, assim, exigir que diferentes informações sobre o tratamento sejam coletadas.
2. **Requisitos emergentes:** que surgem à medida que a compreensão do cliente e dos desenvolvedores cresce durante o desenvolvimento.

Classificação dos requisitos voláteis (cont.)

3. **Requisitos conseqüentes:** que resultam da introdução do sistema nas organização. Pode modificar os processos e criar novos meios de trabalho, que geram novos requisitos de sistema.
4. **Requisitos de compatibilidade:** que dependem de sistemas ou processos de negócio específicos dentro da organização. À medida que eles se modificam, os requisitos de compatibilidade nos sistema encomendado ou entregue também podem ter que evoluir.

Planejamento do gerenciamento de requisitos

1. **Identificação dos requisitos:** identificado de modo único, para referência cruzada entre ele e os outros requisitos e para que possa ser usado na avaliação de facilidade de rastreamento.
2. **Processo de gerenciamento de mudanças:** conjunto de atividades que avalia o impacto e o custo de mudanças.
3. **Políticas de facilidade de rastreamento:** definem as relações entre os requisitos que devem ser registrados e como esses registros devem ser mantidos. Três tipos de informação:

Planejamento do gerenciamento ...(cont.)

- (a) Facilidade de rastreamento de origem: vinculam requisitos aos stakeholders que propuseram esses requisitos (quando uma mudança é proposta, fácil descobrir os stakeholders e consultá-los).
- (b) Facilidade de rastreamento de requisitos: vinculam requisitos dependentes dentro de seu respectivo documento (avaliar quantos requisitos serão afetados por uma mudança e consequência nos outros requisitos).

- (c) Facilidade de rastreamento de projeto: vinculam os requisitos aos módulos de projeto em que esses requisitos são implementados (avalia impacto das mudanças no projeto e implementação).

Planejamento do gerenciamento ...(cont.)

4. **Suporte de ferramentas CASE:** vão desde sistemas especializados de gerenciamento de requisitos a planilhas de cálculo e sistemas simples de bancos de dados. Apoio necessário para:
- (a) armazenamento de requisitos; (b) gerenciamento de mudanças; e (c) gerenciamento de facilidade de rastreamento.

Para sistemas pequenos: recursos de processadores de textos, planilhas de cálculos e bancos de dados.

Gerenciamento de mudanças nos requisitos

1. **Análise do problema e especificação da mudança:** começa com a identificação do problema com os requisitos ou com uma proposta específica de mudança. Análise do problema para verificar validade. Proposta mais específica de mudança pode ser feita.
2. **Análise e custo da mudança:** o efeito é avaliado, com informações sobre facilidade de rastreamento e conhecimento geral dos requisitos. Custo em termos de modificações no documento de requisitos e, quando apropriado, no projeto e implementação. Decisão sobre prosseguir com a alteração ou não.

3. **Implementação de mudanças:** documento de requisito e, quando apropriado, projeto e implementação. Documento deve acomodar mudanças sem muito esforço (minimizar referências externas e seções do documento modulares).
- **Mudanças urgentes:** tentação de fazer a mudança primeiro no sistema e depois no documento de requisitos.
 - **Conseqüência:** especificação de requisitos e implementação não compatíveis.