

EXTRAÇÃO DE REQUISITOS

Requisito

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

Requisitos de software

- requisitos que o produto a ser desenvolvido deve possuir.

Tipos de Requisitos

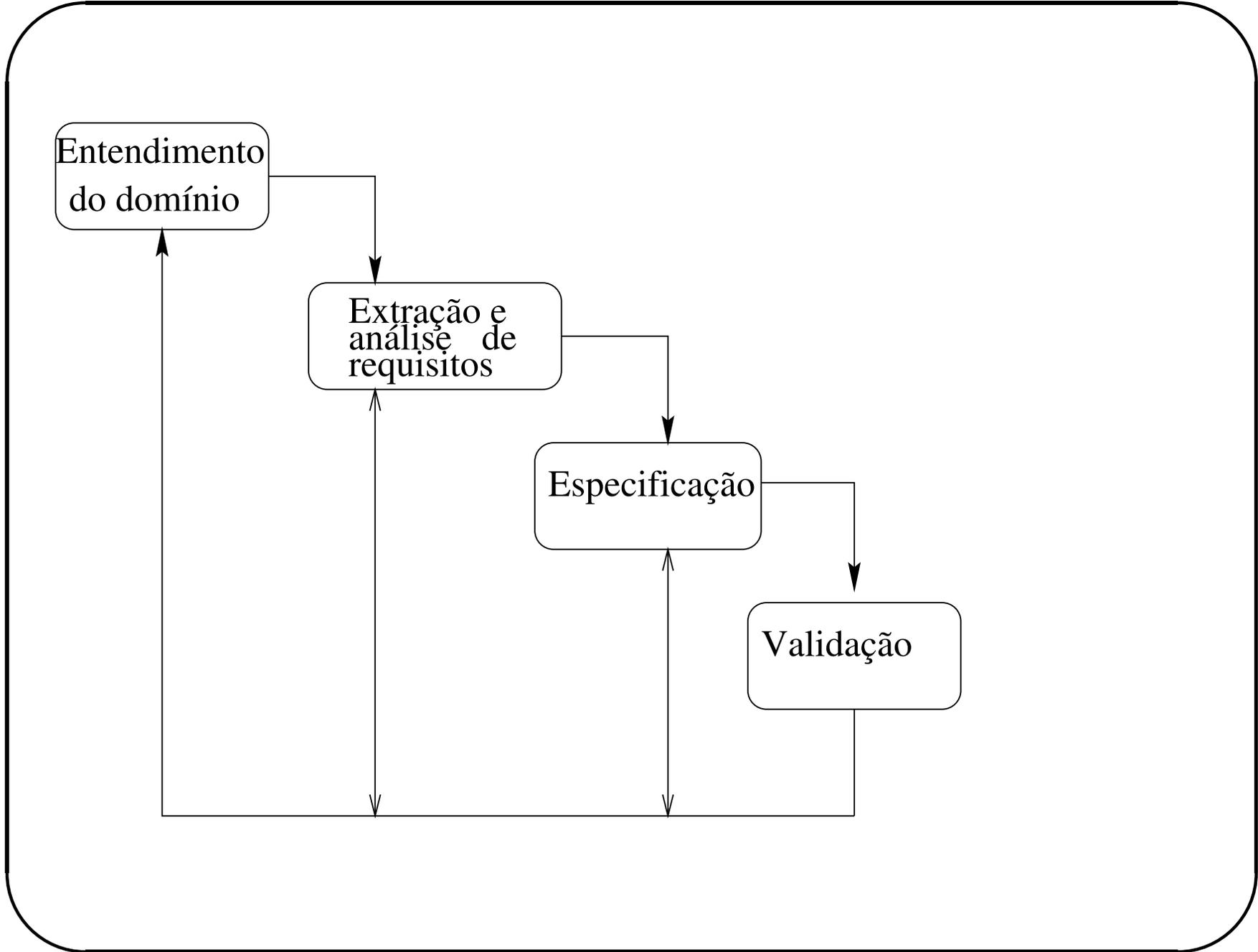
- **Requisitos funcionais:** diretamente ligados à funcionalidade do software.
Exemplo: O sistema deve prever um formulário de entrada dos resultados dos testes clínicos do paciente.
- **Requisitos não funcionais:** expressam qualidades e restrições do software. Exemplos:
 - (a) Dependendo do resultado 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.

Extração de requisitos

- Processo de transformação das idéias que estão na mente dos usuários (a entrada) em um documento formal (saída) através:
 - da determinação dos objetivos do produto; e
 - das restrições para a sua operacionalidade.
- Envolve:
 - análise do problema;
 - documentação dos resultados; e
 - verificação do entendimento do problema.

- saída: *documento de especificação dos requisitos*, que descreve *o que* o produto deverá fazer, e não *como* deve ser feito.
- idealmente: completo e consistente; entretanto, a entrada para esse processo não tem nenhuma dessas propriedades.
- consequência: o processo de extração de requisitos não pode ser totalmente formal e, portanto, totalmente automatizado.

- foco: entendimento do produto a ser desenvolvido e de seus requisitos; quanto mais complexo for o produto, mais difícil se torna o processo.
- princípio da decomposição: ajuda a lidar com a complexidade.



O processo de extração de requisitos

- *entendimento do domínio*: documentos, livros, sistemas, pessoas;
- *extração 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.

- *especificação dos requisitos*: armazenamento dos requisitos em uma ou mais formas, incluindo língua natural, linguagem semiformal ou formal, representações simbólicas ou gráficas;
- *validação dos requisitos*: verificação dos requisitos, visando determinar se estão completos e condizentes com as necessidades e desejos do usuário.

- o processo não é uma seqüência linear dessas atividades
→ elas não podem ser totalmente separadas e executadas seqüencialmente.
- todas são intercaladas e executadas repetidamente.
- no mundo real existe uma sobreposição e um *feedback* entre as atividades.
- algumas partes do produto podem ser analisadas e especificadas enquanto outras ainda estão sendo analisadas.

- a fase de validação pode revelar problemas com a especificação → retorno à fase de análise e especificação.
- problemas com o entendimento do domínio exigem um retorno a essa atividade.
- as necessidades do usuário mudam à medida que o ambiente no qual o sistema funciona muda.
- o próprio documento de especificação de requisitos e o processo de extração dão novas idéias aos usuários sobre as suas necessidades e sobre as funções do sistema.

- PORTANTO: 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.

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 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.

DIFICULDADES DO PROCESSO DE EXTRAÇÃO

1. Falta de conhecimento

- usuário não conhece suas reais necessidades e o que o produto de software pode lhe oferecer.
- desenvolvedores não conhecem o domínio do problema.
- diferenças entre o que os usuários *querem* e o que *precisam*.
- escolhas quando dois requisitos não podem ser totalmente satisfeitos.

2. Problemas de comportamento

- domínio do processo de extração de requisitos pelos desenvolvedores.
- conflitos e ambigüidades nos papéis → clima de insatisfação e participação menos efetiva.
- **RESULTADO:** custo maior, atraso no planejamento e projetos cancelados.

3. Problemas de comunicação

- usuários podem ser incapazes de expressar suas necessidades apropriadamente.
- mal-entendidos entre usuários e desenvolvedores:
 - “implementação” → a escrita do código ou a colocação do produto em funcionamento?
 - “Se a temperatura ou pressão do paciente lida pelo terminal se tornarem *críticas*”.
- ambigüidade da língua natural.

4. Problemas técnicos

- avanço tecnológico muito rápido → problemas cada vez mais complexos e conhecimentos cada vez mais detalhados; requisitos caros ou complexos podem se tornar viáveis.
- alteração nos requisitos.
- muitas fontes de requisitos.

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. o terminal ativará um aviso sonoro; e/ou
 2. o computador principal ativará um aviso sonoro.

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 se a pressão medida está dentro dos parâmetros considerados normais para o paciente (valores próximos ao máximo e mínimo são permitidos).

- (a) valores possíveis para máximo e mínimo?
- (b) máximo < mínimo?
- (c) intervalo fora de um valor normal?
- (d) o que significa valores próximos?

Dificuldade do usuário tomar decisões

- quais as conseqüências das decisões?
- quais as alternativas possíveis?
- necessidades ou perspectivas diferentes:
 - usuários se preocupam com usabilidade e confiança.
 - desenvolvedores com utilização de recursos, algoritmos etc.

Características dos requisitos

- a *frequência*
- a *previsibilidade* da solicitação
- a *atualização* da informação

Frequência

- *Programada*: Desejo receber diariamente uma lista dos pedidos feitos no dia anterior. O relatório deve conter os pedidos efetuados até as 16:30 horas e deve ser entregue às 9 horas da manhã do dia seguinte.
- *Disparada por evento*: Todas as vezes que o nível de estoque de um item ficar abaixo do nível de segurança, forneça um relatório do status do item. O relatório deve ser gerado até a manhã do dia seguinte.
- *Eventual*: Qual é o valor do pedido de compra número 34923? O fornecedor precisa de confirmação agora.

Previsibilidade

- *Previsíveis*

Programada → Desejo receber diariamente uma lista dos pedidos feitos no dia anterior. O relatório deve conter os pedidos efetuados até as 16:30 horas e deve ser entregue às 9 horas da manhã do dia seguinte.

Eventual → Qual é o valor do pedido de compra número 34923? O fornecedor precisa de confirmação agora.

- *Imprevisíveis*

Qual o montante de negócios gerados pelo cliente 34923 nos últimos três meses, até sexta-feira passada?
Terei uma reunião com ele depois de amanhã.

Atualidade dos dados

- *Imediata*: Dados atualizados a cada transação.
Qual é o saldo devedor do cliente da conta número 34923?
- *Adiada*: Dados atualizados ao final do período.
Todas as vezes que o nível de estoque de um item ficar abaixo do nível de segurança, forneça um relatório do status do item. O relatório deve ser gerado até a manhã do dia seguinte.

Complexidade versus custo de processamento

- quando a solicitação é imprevisível e o resultado da informação precisa ser imediato → custo de processamento maior.
- quando a solicitação é previsível e o resultado não precisa ser imediato → custo de processamento menor.

Técnicas de extração de requisitos

- depende da complexidade e dos objetivos do produto de software a ser desenvolvido.
- o desenvolvedor (ou engenheiro de requisitos) → responsável pela produção dos requisitos e lidera o processo.
- conhecimento imprescindível → habilidade de empregar um processo sistemático na extração de requisitos.
- freqüentemente auxiliado por outros desenvolvedores de software, especialistas em documentação e pessoal de apoio.

- os usuários potenciais do produto fazem parte do processo.
- Exemplos:
 1. um “novo e melhor” processador de textos.
 2. produto sem precedentes (pesquisa de mercado, protótipos, testes e avaliações com os usuários).
- **Entretanto:** nenhuma técnica por si só é suficiente para projetos reais.

Procedimentos

- *Perguntar*: identificar a pessoa apropriada e perguntar.
- *Observar e inferir*: observar o comportamento dos usuários e inferir suas necessidades.
- *Discutir e formular*: discutir com os usuários suas necessidades e, juntamente com eles, formular um entendimento comum dos requisitos.

- *Negociar*: começar com um conjunto-padrão e negociar quais desses requisitos serão incluídos, excluídos ou modificados.
- *Estudar e identificar problemas*: identificar os requisitos que podem melhorar o produto
- *Supor*: quando não existe usuário, ou na criação de um produto existente é preciso usar intuição.

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

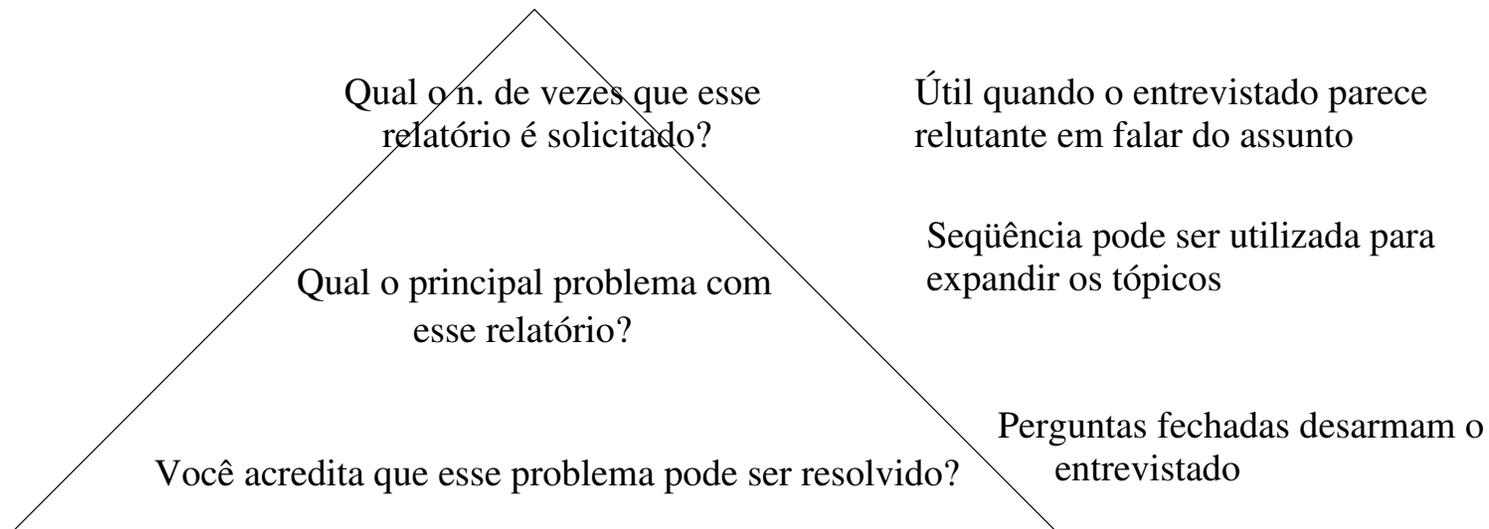
seqüência: dá continuidade a uma questão. “Por que? Dê um exemplo.”

Estrutura da entrevista

- **Pirâmide**

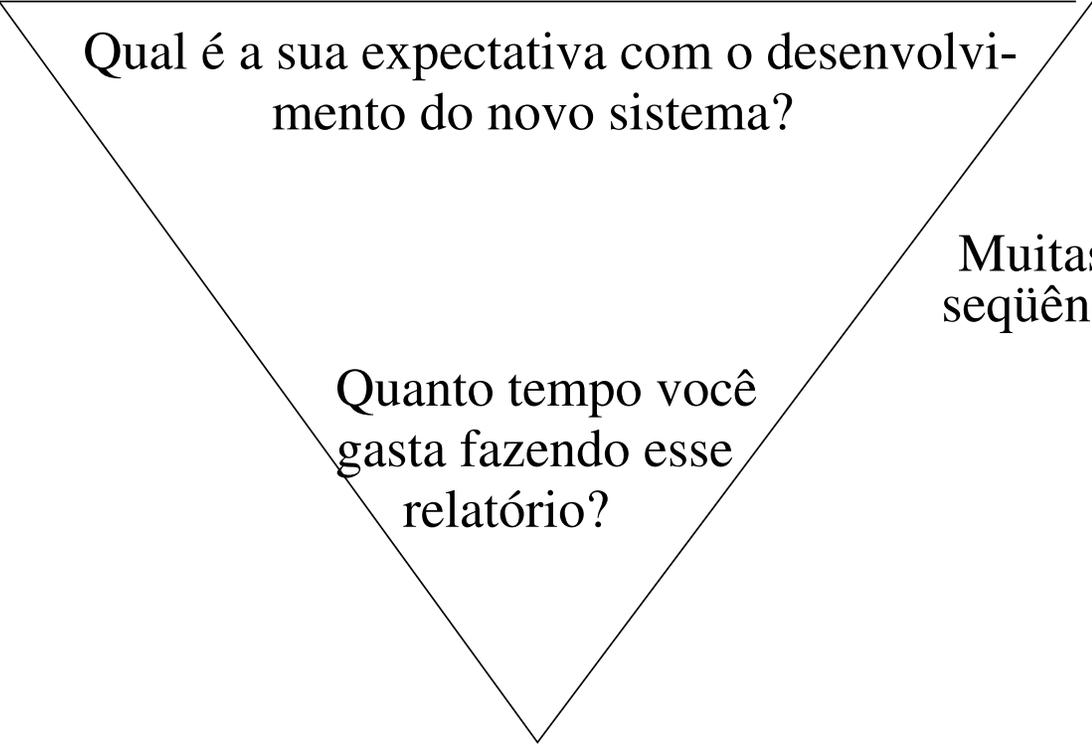
começa com questões fechadas → obtém respostas diretas

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

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

A entrevista fica menos
cansativa pois varia o
tipo de questão

Qual é o n. de vezes que esse relatório é solicitado?

Você acredita que esse problema
pode ser resolvido?

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 sumariar 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

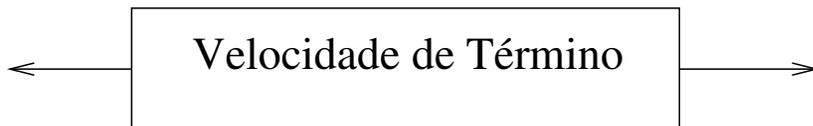
- **Questões abertas-dirigidas:** ‘Por que você acha que os manuais do usuário para o sistema de contabilidade não funcionam?’
 - antecipar o tipo de resposta (enumerá-las)
 - deve ser possível interpretar corretamente as respostas
 - utilizadas quando não é possível listar todas as alternativas

- **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



Rápida

Alta



Baixa

Alta



Baixa

Fácil



Difícil

Difícil



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%

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 restringi-la.

- especialmente útil no começo do processo de extração de requisitos pois:
 - a 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).
- o 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.

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

- os 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 são sempre importantes.
- 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.

JAD

- técnica para promover cooperação, entendimento e trabalho em grupo entre usuários e desenvolvedores.
- facilita a criação de uma visão compartilhada do que o produto de software deve ser.
- desenvolvedores ajudam usuários a formular problemas e explorar soluções.
- usuários ganham um sentimento de envolvimento, posse e responsabilidade para com o sucesso do produto.

- quatro princípios:

1. *dinâmica de grupo*, com a utilização de sessões de grupo facilitadas para aumentar a capacidade dos indivíduos;
2. *uso de técnicas visuais* para aumentar a comunicação e o entendimento;
3. *manutenção do processo organizado e racional*;
4. *utilização de documentação-padrão*, preenchida e assinada por todos os participantes.

- duas etapas principais:
 1. planejamento: extração e especificação de requisitos.
 2. projeto de software.
- cada etapa consiste de três fases:
 1. *adaptação*;
 2. *sessão*; e
 3. *finalização*.

- **Adaptação:** preparação para a sessão:
 - organizar a equipe;
 - adaptar o processo JAD ao produto a ser construído;
 - preparar o material.
- **Sessão:** um ou mais encontros estruturados, envolvendo desenvolvedores e usuários:
 - requisitos são desenvolvidos e documentados.
- **Finalização:** converter a informação da fase de sessão em sua forma final, em um documento de especificação de requisitos.

- Seis componentes:

1. *líder da sessão;*

2. *engenheiro de requisitos;*

3. *executor;*

4. *representantes dos usuários;*

5. *representantes de produtos de software; e*

6. *especialista*

Líder da sessão

- responsável pelo sucesso do esforço e facilitador dos encontros:
 - familiaridade com todos os aspectos do JAD;
 - habilidade para gerenciar encontros;
 - experiência na área da aplicação (planejar e entender as várias tarefas da técnica e saídas);

- bom relacionamento pessoal e qualidades de liderança para:
 - entender e facilitar a dinâmica de grupo;
 - iniciar e manter o foco das discussões;
 - reconhecer quando os encontros estão saindo da meta e trazê-los de volta para o objetivo;
 - lidar eficientemente com personalidades e comportamentos diferentes dos participantes; e
 - permanecer entusiasmado ao longo de encontros demorados e difíceis.

Engenheiro de requisitos

- participante diretamente responsável pela produção dos documentos de saída das sessões JAD:
 - desenvolvedor experiente para entender questões técnicas e detalhes discutidos durante as sessões.
 - habilidade de organizar idéias e expressá-las com clareza.
 - saber usar as ferramentas de software necessárias (produção de documentos ou ferramentas de prototipagem).

Executor

- responsável pelo produto sendo construído, com duas principais responsabilidades:
 1. dar aos outros participantes uma visão dos pontos estratégicos do produto a ser construído (o porquê de ele estar sendo construído e como a empresa espera melhorar com a utilização do novo produto).
 2. tomar decisões executivas (alocação de recursos, que podem afetar os requisitos e o projeto do novo produto).

Representantes dos usuários

- pessoas na empresa que irão utilizar o produto de software:
 - gerentes ou pessoas-chave dentro da empresa (melhor visão do sistema todo e de como ele será usado), selecionados de acordo:
 - * com o conhecimento de suas próprias necessidades dentro da empresa;
 - * o entendimento de como seu departamento interage com outros departamentos; e
 - * algum conhecimento de produtos de software.

Representantes de produtos de software

- pessoas bastante familiarizadas com as capacidades dos produtos de software, cujo papel é:
 - ajudar os usuários a entender o que é razoável ou possível que o novo produto faça.
 - esclarecer o usuário sobre as tecnologias existentes.
 - entender as conseqüências de escolher um ou outro caminho para resolução do problema.

Especialista

- pode fornecer informações detalhadas sobre um tópico específico:
 - especialista da comunidade de usuários → usa um determinado tipo de relatório, ou é responsável pela execução de um determinado tipo de pedido.
 - especialista da comunidade de desenvolvedores → conhecesse os detalhes da rede interna da empresa (protocolos de comunicação).

A fase de adaptação

- deve ser adaptada a cada produto de software a ser desenvolvido.
- responsabilidade do líder da sessão, com a ajuda de um ou dois desenvolvedores.

1. *Preparação*: o líder da sessão e desenvolvedores devem se inteirar do que foi conseguido, tipo de produto sendo discutido e decisões tomadas.
 - pequenos encontros com um ou mais usuários e talvez um encontro com o executor.
 - o líder da sessão e o engenheiro de requisitos podem necessitar de familiarização com a empresa ou departamento para o qual o produto será construído.
 - organograma da companhia ajuda na identificação das pessoas-chave que realmente irão contribuir.

2. *Organizar o grupo*: líder seleciona os participantes; executor também pode ter identificado alguns.
- líder prepara os participantes para a sessão (data, hora e lugar, uma lista de perguntas).
 - requisitos de alto nível: objetivos, benefícios esperados, restrições, adaptados ao produto a ser desenvolvido.
 - os participantes abordam as questões de acordo com as suas perspectivas (usuários → ponto de vista comercial, representantes de produtos de software → tecnológico).

3. *Ajustar o processo*: líder usa sua experiência e julgamento para adaptar o processo ao produto a ser construído (tempo e quantos encontros necessários, formato geral dos documentos).
4. *Preparar material*: líder faz arranjos logísticos (reserva e organização da sala do encontro).
 - recursos visuais (transparências, canetas de marcação, lousa de papel, computador etc.)
 - mensagem de boas-vindas, agenda, revisão do processo e categorias de requisitos de alto nível, questões sobre escopo do produto e formulários JAD.

A fase de sessão

- um ou mais encontros do grupo para definir os requisitos de alto nível para o novo sistema, assim como seu escopo.
- participantes trazem idéias e visões diferentes do produto para a sessão.
- através de discussões cuidadosas e facilitadas, essas idéias e visões são apresentadas, analisadas e refinadas.

1. *Preparação*: sessão começa com o líder e o executor dando boas-vindas aos participantes.
 - todos os participantes são apresentados.
 - executor faz um breve resumo do esforço feito até o momento e descreve as expectativas dos participantes em relação à sessão.
 - o líder dá uma visão geral do processo, incluindo o tempo a ser gasto em cada tarefa.
 - à medida que uma nova tarefa começa, o líder fornece informações mais detalhadas sobre ela (motivo da tarefa, os papéis dos participantes, como a tarefa é executada e como as saídas são registradas e formatadas).

2. *Definir requisitos de alto nível.* Cinco tópicos:

- 1) Objetivos: finalidade da construção desse produto
- 2) Benefícios esperados: quantificáveis ou não, tangíveis ou intangíveis
- 3) Estratégias e considerações futuras: como esse produto pode ajudar na organização, avanço estratégico ou competitivo?
- 4) Restrições e suposições: recursos, estrutura organizacional, padrões, leis?
- 5) Segurança, auditoria e controle: requisitos de segurança internos ou externos, auditorias ou controles?

3. *Delimitar o escopo do sistema*

- quem realmente vai usar o produto.
- quais as principais funções que o produto ajudará a executar.
- funcionalidades que estão fora do escopo do sistema (delimitar o escopo).

4. *Documentar questões e considerações*: algumas afetam o processo JAD, outras não, mas podem afetar a maneira como o produto será construído ou utilizado.

5. *Concluir a fase de sessão*: revisão da informação coletada e das decisões tomadas.

- cada participante tem a oportunidade de expressar preocupações sobre os requisitos remanescentes.
- todos adquirem um senso de posse e de responsabilidade para com os requisitos documentados.
- a conclusão da sessão de forma positiva garante contribuições futuras de todos os participantes.

A fase de finalização

- transformar as transparências, anotações da lousa de papel e outros documentos em documentos de especificação dos requisitos de software.
- desenvolvedores trabalham em tempo integral durante essa fase, auxiliados pelo líder da sessão.

- Três etapas distintas:

1. *Completar o documento*: formato fixo para o documento, mas pode ser adaptado.
2. *Revisar o documento*: participantes revisam e fazem comentários substanciais → um novo encontro pode ser marcado.
3. *Obter a aprovação do executor*: todos os participantes recebem uma cópia do documento final.

PROTOTIPAGEM

- comparação com um produto de software que sirva de referência.
- criar um produto que ilustre as características relevantes.
- os usuários podem descobrir quais são as suas reais necessidades.

- processo:
 - um estudo preliminar dos requisitos do usuário; e
 - um processo iterativo de construção do protótipo e avaliação junto dos usuários.
- cada repetição permite que o usuário entenda melhor seus requisitos, inclusive as implicações dos requisitos articulados nas iterações anteriores.
- um conjunto final de requisitos pode ser formulado, e o protótipo descartado.

- benéfica somente se o protótipo puder ser construído substancialmente mais rápido que o sistema real.
- usada para extrair e entender requisitos.
- seguida pelo processo estruturado e gerenciado de construção do sistema propriamente dito.
- pode ser muito útil para superar várias dificuldades de comunicação e de articulação do usuário.

Comentários finais

- ferramentas para auxiliar na extração de requisitos.
- propiciam condições para que as pessoas possam trabalhar juntas, sem que necessariamente estejam na mesma sala ou prédio (videoconferência).
- ferramentas para prototipagem e para produção de documentos.
- ferramentas podem ser úteis na fase de consolidação (edição e a reordenação das idéias).