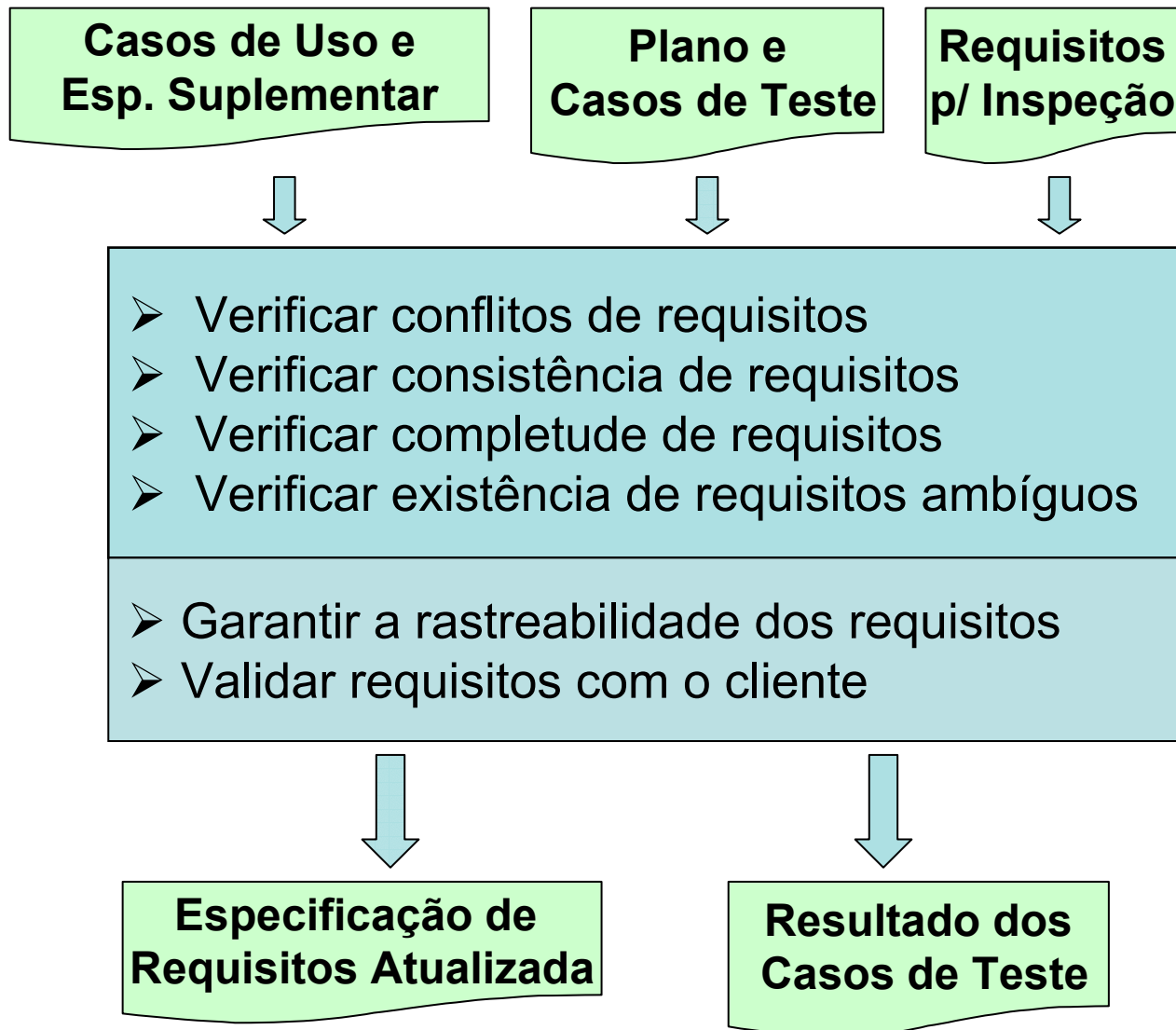


Verificação e Validação de Requisitos

Verificação e Validação dos Requisitos



Inspetor



Analista de Requisitos

Diretrizes para uma boa especificação

1. Separe funcionalidade de implementação
2. É necessária uma linguagem de especificação de sistemas orientada ao processo
3. A especificação deve abranger o sistema do qual o software é um componente
4. Uma especificação deve abranger o ambiente no qual o sistema opera
5. Uma especificação de sistema deve ser um modelo cognitivo
6. Uma especificação deve ser operacional
7. A especificação do sistema deve ser tolerante com a não completude e ser expansível
8. Uma especificação deve ser localizada e fracamente acoplada

(Balzer, Goldman, Wile, 1978)

Revisão da Especificação (nível macroscópico)

- Os revisores tentam garantir que a especificação seja completa, consistente e precisa
- Questões:
 - Metas e objetivos do software permanecem consistentes com metas e objetivos do sistema?
 - O fluxo e a estrutura de informação são adequadamente definidas para o domínio da informação?
 - O modelo de casos de uso estão claros?
 - Foram descritas as interfaces importantes para todos os elementos do sistema?

Revisão da Especificação (nível macroscópico)

- As funções importantes permanecem dentro do escopo e cada uma foi adequadamente descrita?
- O comportamento do software é consistente com a informação que ele deve processar e as funções que deve executar?
- As restrições de projeto são realísticas? Qual é o risco tecnológico de desenvolvimento? Requisitos de software alternativos foram considerados?
- Critérios de Validação foram declarados detalhadamente? Eles são adequados para descrever um sistema bem sucedido?
- Existem inconsistências, omissões ou redundâncias?
- Como as estimativas do Plano de Projeto de Software foram afetadas?

Revisão da Especificação (nível detalhado)

- A preocupação é com o enunciado da especificação. Tenta-se descobrir problemas que possam estar ocultos no conteúdo da especificação
- Diretrizes:
 - Esteja alerta para perceber conectivos persuasivos (certamente, claramente, obviamente) e perguntar por que eles estão presentes
 - Procure termos vagos e peça esclarecimento (algum, às vezes, usualmente, freqüentemente)
 - Quando forem fornecidas listas que não sejam completas, certifique-se de que todos os itens sejam entendidos (evite colocar **etc, tal como, assim por diante**)

Revisão da Especificação (nível detalhado)

- Esteja certo de que os limites declarados não contenham pressuposições não declaradas
 - “os códigos válidos variam de 0 a 100” - números inteiros ou reais?
- Cuidado com verbos vagos. Há muitas maneiras de interpretá-los
 - manuseado, rejeitado, pulado
- Cuidado com pronomes "pendentes"
 - o módulo I/O comunica-se com o módulo de validação de dados e seu sinal de controle está ligado. Sinal de controle de qual dos dois módulos?
- Procure declarações que impliquem certeza (sempre, cada, todos, nunca) e depois peça prova

Revisão da Especificação (nível detalhado)

- Quando um termo for explicitamente definido num lugar, evite utilizar outras definições para o mesmo termo (normalização dos termos: documento - arquivo)
- Quando uma estrutura for descrita em palavras, verifique se há um gráfico ou uma figura para auxiliar a compreensão
- Quando um cálculo for especificado, desenvolva pelo menos dois exemplos

Desenvolvimento orientado a reuso

O que vem depois?

- A especificação de Requisitos e os modelos elaborados na *Análise* são a base para o *design* do sistema
- A fase de *design* vai evoluir os modelos incorporando características de implementação
- Neste momento
 - é importante uma abordagem gerencial para incrementar a produtividade
 - é importante garantir que os modelos do software atendam aos requisitos

Desenvolvimento orientado a reuso

- Baseado em reutilização de componentes de software
- Podem ser acessados com alguma infra-estrutura de integração para estes componentes
- *Design* com reuso:
 - reuso de sistemas de aplicações: COTS
 - reuso de componentes
 - reuso de funções
- Vantagens:
 - reduzir a quantidade de software a ser desenvolvido
 - reduzir o prazo de desenvolvimento
 - reduzir os custos

Sistemas COTS

- COTS (*Commercial off-the-shelf*)
- Produtos de Prateleira: componente oferecido por um terceiro
- Um sistema de aplicações pode ser reutilizado pela sua incorporação, sem mudança, em outros sistemas
- Exemplo:
 - Sistemas de Comércio Eletrônico
 - Base de Dados
 - Compra e Venda de Produtos (carrinho de compra)
 - Processo de Pagamento

Sistemas COTS

- Desvantagens:
 - O produto final pode não ser aquele que o cliente pediu
 - Adequação dos requisitos é inevitável
 - Problemas com interoperabilidade de sistemas COTS
 - Controle fraco sobre a evolução do sistema
 - Suporte técnico dos fabricantes de COTS

Validação dos Requisitos

Validação dos Requisitos

- Será que realmente entendi o que o cliente deseja?
- Devo me certificar de que não houve falha em nossa interação (comunicação)
- Há diversas técnicas de validação

Validação de Requisitos

- Demonstrar que os requisitos definem o sistema que o cliente realmente deseja
- Custos com erros de requisitos são altos
 - Consertar um erro de requisitos após entrega do sistema pode custar mais de 100 vezes o custo de um erro de implementação

Validação de Requisitos

- **Validade:** O sistema possui as funções para suprir as necessidades dos usuários?
- **Completude:** Foram incluídas todas as funções requisitadas pelo cliente?
- **Consistência:** Existe algum requisito conflitante?
- **Não ambíguos:** Todos estão descritos de forma clara e objetiva?
- **Verificação:** Os requisitos podem ser verificados?
- **Rastreáveis:** os requisitos tem definidos:
 - A origem?
 - As interdependências entre os requisitos?
 - Os relacionamentos com os diagramas de projeto e componentes de implementação?

Técnicas de Validação de Requisitos

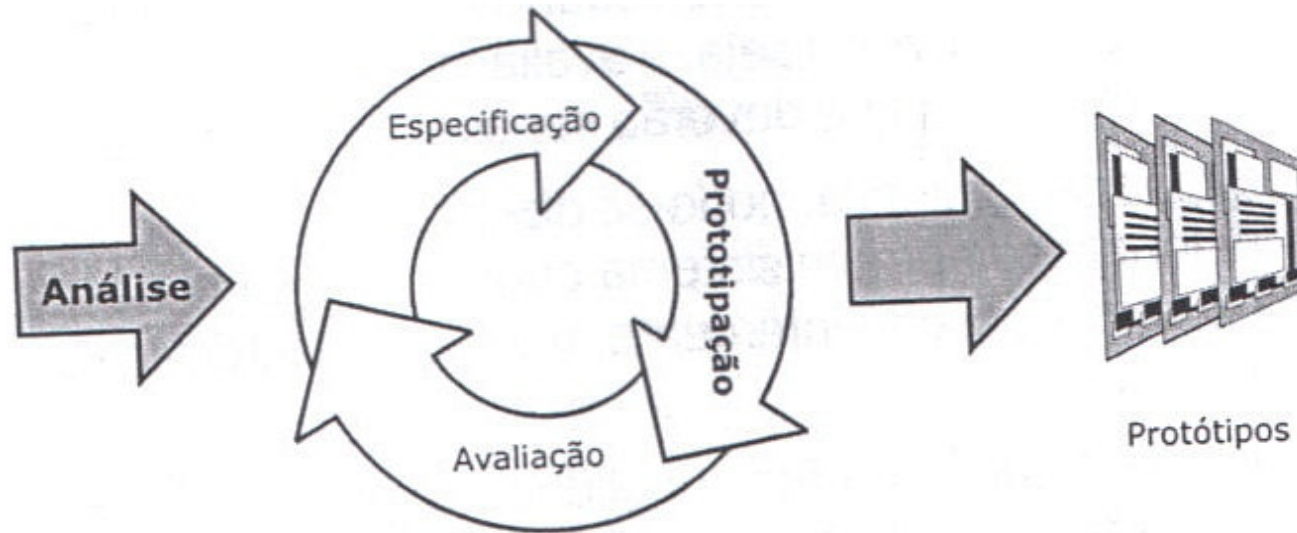
- Revisões de Requisitos
 - Análise manual sistemática dos requisitos
- Prototipação
 - Uso de modelo “executável” (interfaces) do sistema para avaliar requisitos
- Geração de Casos de Teste
 - Desenvolver testes específicos para avaliar os requisitos
 - Análise de Consistência Automática
 - Avaliar uma especificação dos requisitos

Processo de Projeto de Sistemas - Prototipação

Processo de Projeto de Sistemas - Prototipação

- Uma forma de avaliação de interface é a prototipação
- Provê a avaliação das interfaces junto aos clientes, com o auxílio de técnicas apropriadas (usabilidade)
 - A partir desta avaliação um novo ciclo de especificação, prototipação e avaliação deve ser realizada

Processo de Projeto de Interfaces



Prototipação

- O protótipo permite ao projetista avaliar seu projeto ao longo do processo de criação
- A prototipação é parte essencial do processo de projeto de interface

Prototipação

- O esforço envolvido na especificação, no projeto e na implementação de uma interface com o usuário representa parte significativa dos custos de desenvolvimento de aplicações
- Como este é um artefato não-acabado e com finalidade de testes, a sua construção deve ser rápida e de baixo custo

Ferramenta para especificar e gerenciar requisitos

Ferramentas de Especificação Automatizadas

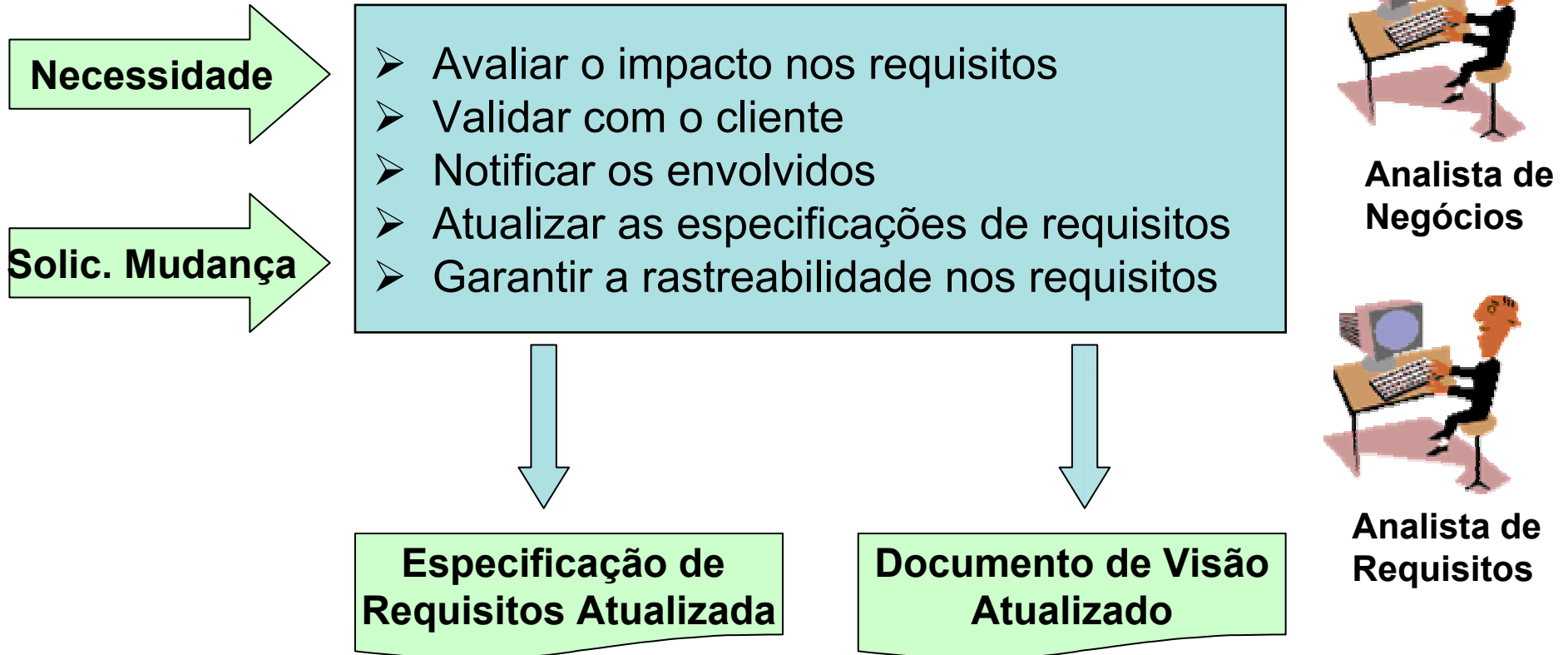
- 1ª categoria:
 - técnicas automatizadas que nada mais são do que um método manual que foi complementado com uma ferramenta CASE
 - possibilitam que o analista atualize informações e rastreie as conexões entre representações novas e existentes do sistema
 - Ex: *RequisitePro* (IBM)

Ferramentas de Especificação Automatizadas

- 2ª categoria:
 - técnicas automatizadas que fazem uso de uma notação especial (na maioria dos casos, essa é uma linguagem de especificação de requisitos) que foi explicitamente projetada para processamento usando-se uma ferramenta automatizada
 - Ex: PSL/PSA (linguagem de especificação: PSL)

Rastreabilidade e Gestão de Mudanças

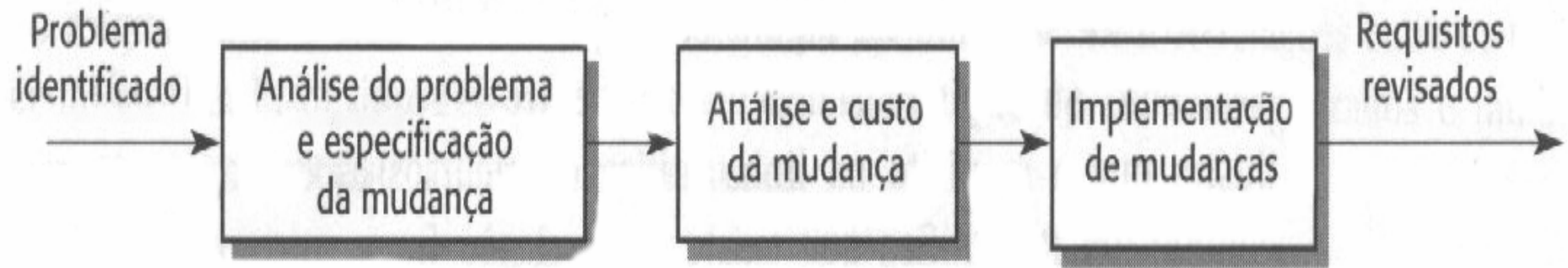
Rastreabilidade e Gestão de Mudanças



Gerência de Mudanças

- Requisitos são inevitavelmente incompletos e inconsistentes
 - Requisitos novos surgem durante o processo
 - mudanças nas necessidades do negócio
 - melhor entendimento do sistema que está sendo desenvolvido
 - Diferentes pontos de vista têm diferentes requisitos e esses geralmente são contraditórios
 - É função do analista durante a elicitação de requisitos identificar
 - Requisitos contraditórios
 - Tendências de mudanças

Processo de Gerência de Mudanças



O que é Baseline ?

- Baseline – linha base
- Uma *baseline* é uma 'imagem' de uma versão de cada artefato no repositório do projeto
- Funciona como um padrão oficial básico para os trabalhos subseqüentes
- Somente mudanças autorizadas podem ser efetuadas na *baseline*
- *Uma baseline de requisitos é uma versão da especificação de requisitos*
 - Todo o conjunto

Gerência de Mudanças

- O Gerenciamento de Mudanças envolve:
 - a identificação da baseline de requisitos
 - a restrição de mudanças
 - a auditoria das mudanças
 - Que mudanças já foram efetuadas?
 - Por que?
 - Quais os problemas?
- Uma mudança nos requisitos pode implicar em alterações em um ou mais modelos subsequentes do software

Gerência de Mudanças

- Mudança
 - Terá que ser efetuado um planejamento para acomodação da mudança
 - Custo
 - Prazo
 - Revisar requisitos para evitar a introdução de conflitos
 - Questionar stakeholders que especificaram um requisito sendo alterado para obter concordância com a alteração

Rastreabilidade

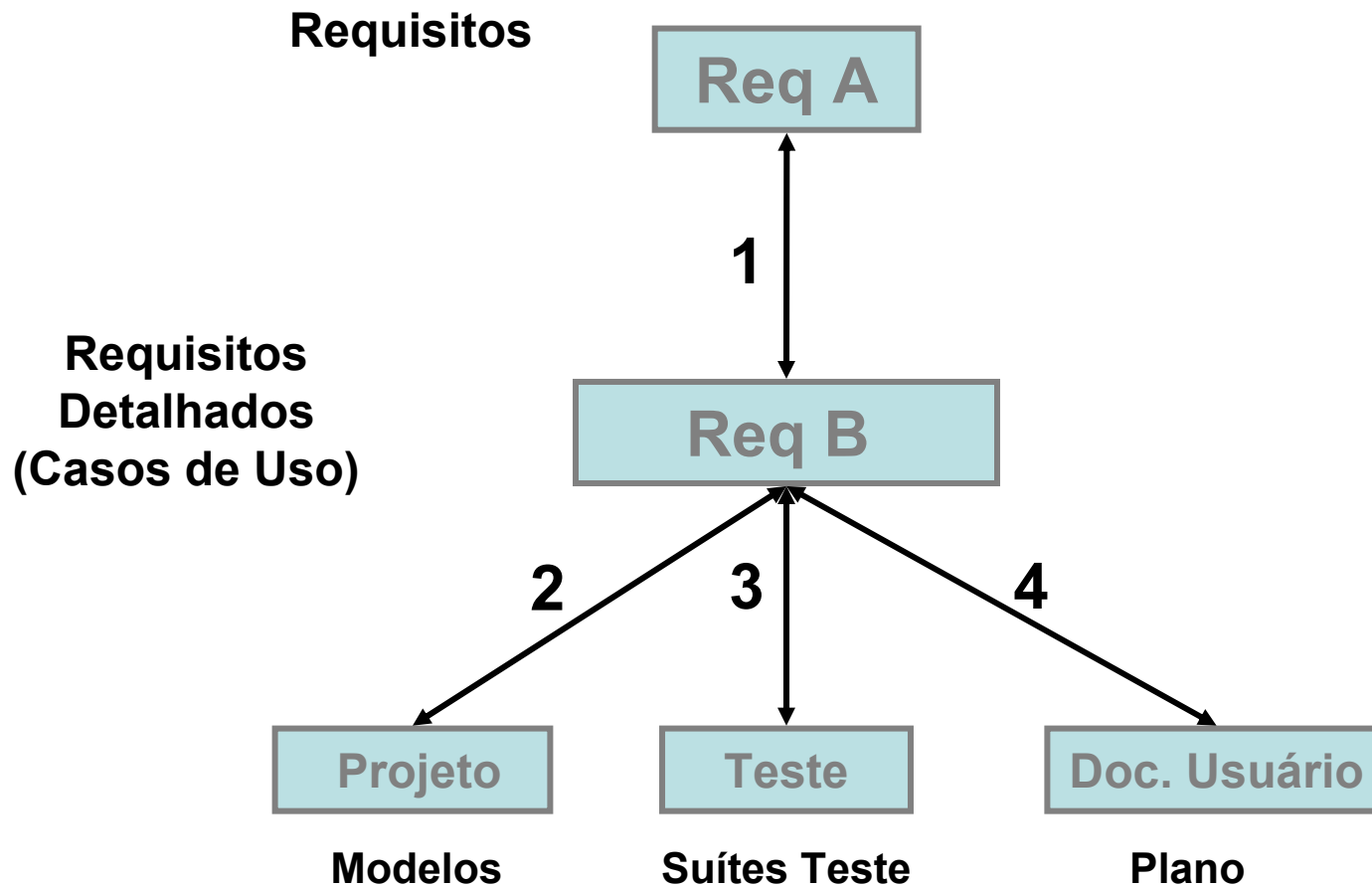
Rastreabilidade

- Responsável por dependências entre requisitos, suas origens e projeto do sistema
- Rastreamento de Origem
 - Associação entre requisitos e *stakeholders* que propuseram tais requisitos

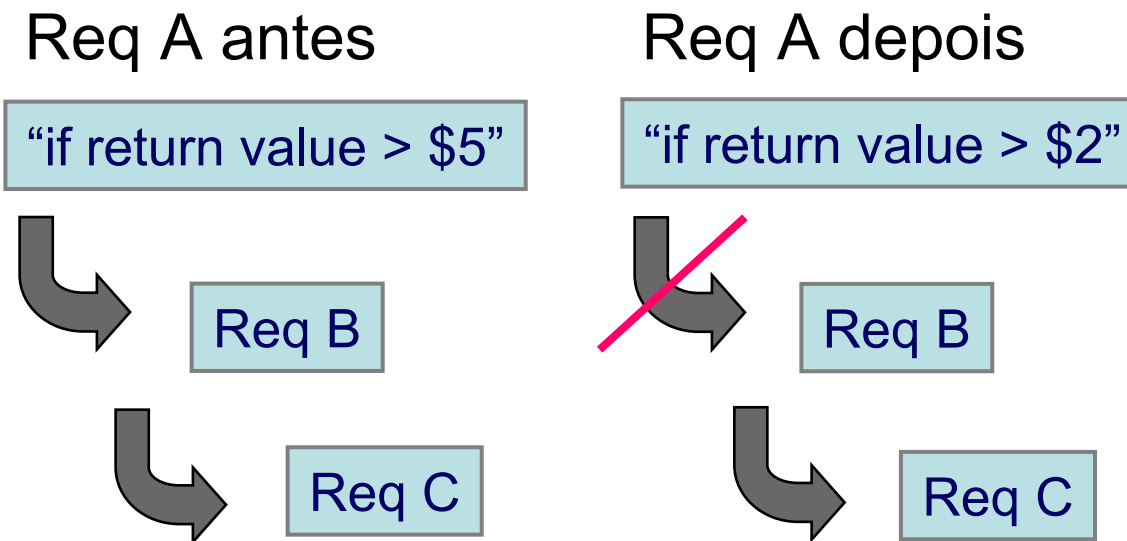
Rastreabilidade

- Rastreamento de Requisitos
 - Associação entre requisitos dependentes
- Rastreamento de Projeto
 - Associação dos requisitos com o projeto
- Usar hipertexto ou referência cruzada
 - Ou matriz de rastreamento

Rastreabilidade



Rastreabilidade: Análise de Impacto



- Links dos requisitos devem ser marcados como “suspeitos”
- Links “suspeitos” devem ser analisados

Gerência de Escopo

Gerência de escopo

- O escopo de um projeto é definido pelo conjunto de requisitos alocados para ele
- Para o projeto se adequar aos recursos disponíveis (tempo, pessoas e dinheiro) é primordial o gerenciamento de escopo com êxito

Gerência de escopo

- Inclui certificar-se que o projeto não crescerá além dos:
 - Requisitos necessários
 - Orçamento planejado
 - Prazo estabelecido

Gerência de escopo

- É feito com o detalhamento do fluxo de trabalho com a finalidade de:
 - Priorizar e refinar as informações fornecidas para selecionar as características e os requisitos que serão incluídos
 - Listar o conjunto de casos de uso (ou cenários) que representam alguma funcionalidade central e significativa
 - Definir quais atributos dos requisitos e rastreabilidades devem ser mantidas

Referências Bibliográficas ^{1/2}

- Balzer, R.M.; Goldman, N.;Wile, D. *Informality in program specifications*. IEEE Transactions on Software Engineering, Ney York, V.SE-4, p.94-103, 1978.
- Breitman, K.; Sayão, M. Gerência de Requisitos. Simpósio Brasileiro de Engenharia de Software - SBES, 2005.
- Delicato, F. Modelagem de Requisitos. Notas de Aula. RN L UFRN. 2006.
- Engenharia de Requisitos. RJ: PUC-Rio. http://www.maxwell.lambda.ele.puc-rio.br/cgi-bin/PRG_0599.EXE/6954_3.PDF?NrOcoSis=19742&CdLinPrg=en
- Fortes, R.P.M. Capítulo 6 Princípios Fundamentais da Análise de Requisitos Engenharia de Software – Pressman. Notas de aula. SP : USP, 2006.
- Leite, J. III. Requisitos são Frases. Notas de aula. RJ : Puc-Rio. 1997.
- Mota, A. Engenharia de Requisitos. Notas de aula. PE : UFPE. 2006.
- PETROBRAS, Documento de Gerenciamento de Requisitos. PI-PR-11-00132-0. TI/IDTA, 2006.

Referências Bibliográficas ^{2/2}

- Processos da Engenharia de Requisitos: elicitação e análise de requisitos. Notas de aula. BA : UFBA, 2005
- Sanchez, M.L. Modelagem Semi-formal de Sistemas: Orientação a Objetos. Notas de Aula. UFF, 2006.
- Sommerville, I. Engenharia de Software, São Paulo: Pearson Addison Wesley, 2003.
- Techniques for Requirements Elicitation. <http://www-di.inf.puc-rio.br/~karin//pos/goguen.pdf>
- Valacich, J *et al.* *Essential of systems Analysis & Design*. New Jersey : Pearson Education, 2001.