SSC721 – Teste e Inspeção de Software

Inspeção de Software – DR

Profa. Ellen Francine Barbosa
francine@icmc.usp.br

Instituto de Ciências Matemáticas e de Computação — ICMC/USP

Roteiro

- Inspeção em DR
- Conclusão
- Exercício de Fixação

Aula Anterior...

Revisões de Software

- Discussão informal de um problema técnico.
- Apresentação do projeto de software para uma audiência de clientes, administradores e pessoal técnico.
- Revisões Técnicas Formais (RTF), as quais incluem avaliações técnicas do software realizadas em pequenos grupos.
  - Inspeção
  - Walkthrough
  - Peer-Review

Aula Anterior...

Inspeção de Software

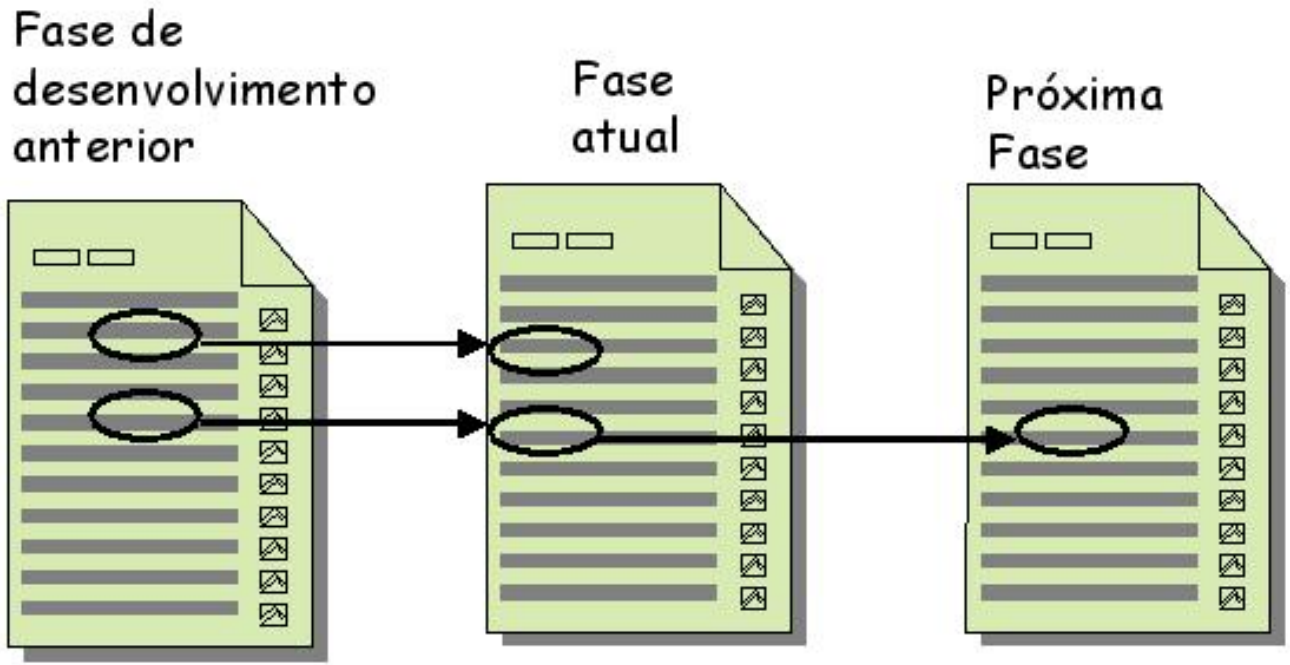
- Método de análise estática para verificar a qualidade de um produto de software.
- Pode-se inspecionar tanto produtos de software como também projetos de software.
  - Diferencial está na seleção dos aspectos que devem ser considerados durante a revisão.
- **Inspeção em Código-fonte**.
- **Inspeção em Documentos de Requisitos**.

Inspeção em Documento de Requisitos

- Visa a encontrar defeitos em documento de requisitos, realizando uma análise estática.
- *Mas... que tipo de defeito???*

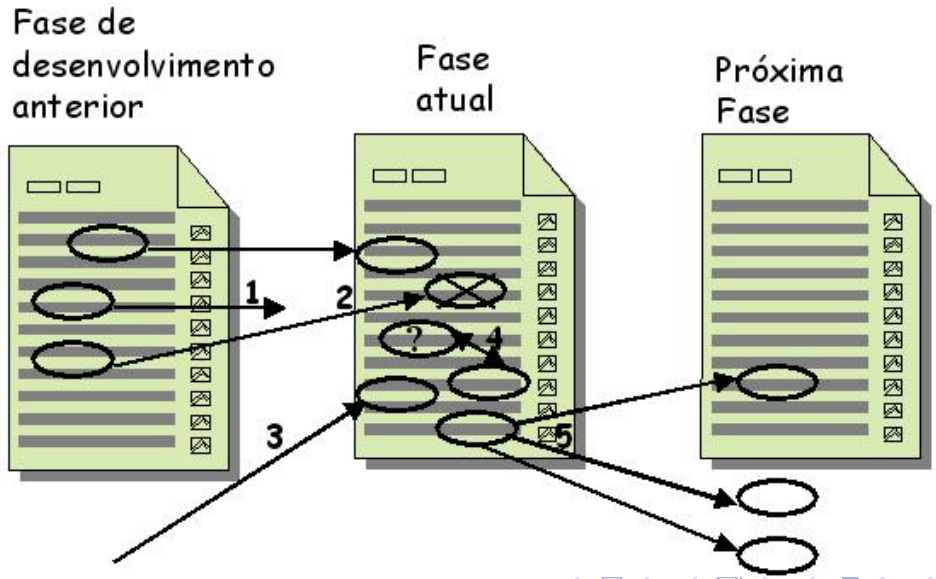
Taxonomia de Defeitos em Documento de Requisitos I

- Utilizada para classificação dos defeitos encontrados.
- Situação ideal:
  - A informação é transformada corretamente.



Taxonomia de Defeitos em Documento de Requisitos II

- Tipos de erros:
  1. A informação é perdida durante a transformação.
  2. A informação é transformada incorretamente.
  3. Informação estranha é introduzida.
  4. A mesma informação é transformada em diversas ocorrências inconsistentes.
  5. A mesma informação possibilita diversas transformações inconsistentes.



Taxonomia de Defeitos em Documento de Requisitos III

- Classes de Defeitos:
  - Defeitos de Omissão
  - Defeitos de Fato Incorreto
  - Defeitos de Inconsitência
  - Defeitos de Ambigüidade
  - Defeitos de Informação Estranha

Defeitos de Omissão I

- Defeito de Omissão (O): qualquer informação necessária que tenha sido omitida.
- Exemplo 1 (Omissão de Funcionalidade). Considere um sistema de biblioteca e os seguintes requisitos funcionais (RF):
  - RF2: O sistema deve solicitar a informação necessária para inserir um item bibliográfico: título, autor, data, lugar, assunto, resumo, número, editor, periódico, congresso.
  - RF3: O sistema deve dar uma mensagem de alerta quando o usuário tentar inserir um item incompleto. Essa mensagem deve questionar o usuário se ele deseja cancelar a operação, completar a informação ou concluir a inserção como está.

    Qual informação é necessária para possibilitar uma inserção incompleta?

Defeitos de Omissão II

- Exemplo 2 (Omissão de Desempenho:)
  - RNF1: O sistema deve fornecer os resultados tão rápido quanto possível.

Tão rápido quanto possível?

Defeito de Fato Incorreto

- Defeito de Fato Incorreto (FI): informação que consta do artefato mas que seja contraditória com o conhecimento que se tem do domínio de aplicação.
- Exemplo: Considere um sistema de empréstimo numa biblioteca e o seguinte RF:
  - RF30: O sistema não deve aceitar devolução de livros se o usuário não estiver com a carteirinha da biblioteca no momento.

Para devolução de livros não é necessário apresentar a carteirinha pois todas as informações já estão registradas no sistema !

Defeito de Inconsistência

- Defeito de Inconsistência (I): informação que consta do artefato mais de uma vez e em cada ocorrência ela é descrita de forma diferente.
- Exemplo: Considere um sistema de empréstimo numa biblioteca e o seguinte RF:
  - RF5: O sistema não deve permitir períodos de empréstimo maiores que 15 dias.
  - RF9: Professores podem retirar livros por um período de 3 semanas.

Qual período deve ser considerado?

Defeito de Ambigüidade

- Defeito de Ambigüidade (A): quando a informação pode levar a múltiplas interpretações.
- Exemplo: Considere um sistema de empréstimo numa biblioteca e o seguinte RF:
  - RF20: Se o número de dias que o usuário está em atraso é menor que uma semana, ele deve pagar uma taxa de R$ 1,00; se o número é maior que uma semana, a taxa é de R$ 0,50 por dia.

Qual a taxa a ser paga se o período for de uma semana?
No primeiro caso, a taxa deve ser calculada por dia?

Defeito de Informação Estranha

- Defeito de Informação Estranha (IE): qualquer informação que, embora relacionada ao domínio, não é necessária para o sistema em questão.
- Exemplo:
  - RF15: Quando um novo livro é adicionado ao acervo, ele permanece em uma prateleira especial por um período de um mês.

Essa informação não é necessária ao sistema !!

Inspeção em Documento de Requisitos

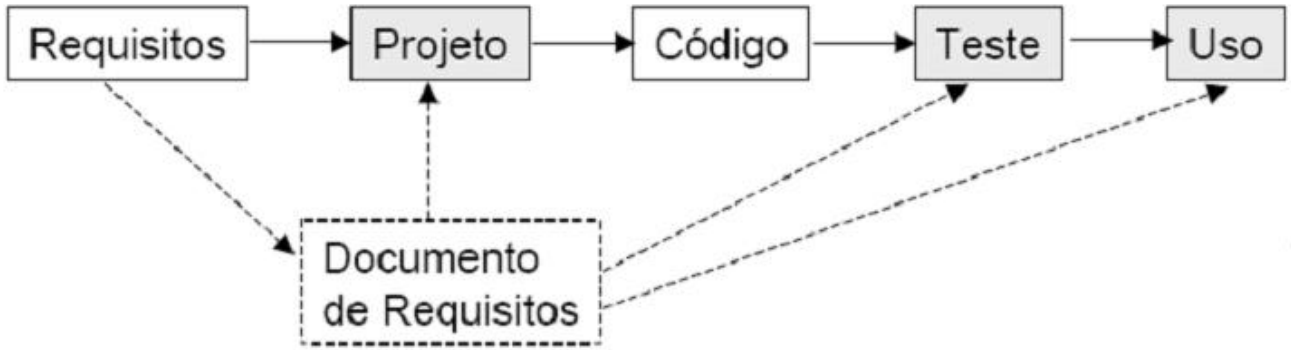
- Visa a encontrar defeitos em documento de requisitos, realizando uma análise estática.
- *Mas... que tipo de defeito???*
- *E... como encontrá-lo???*

Leitura Baseada em Perspectiva (PBR) I

- Técnica de leitura proposta para detectar defeitos em documentos de requisitos.
- Faz com que cada revisor se torne responsável por uma perspectiva em particular.
- Requer que o revisor crie abstrações de um produto de software e responda algumas questões a partir da análise da abstração.
  - As questões a serem respondidas e as regras para criar a abstração são definidas para cada perspectiva diferente.

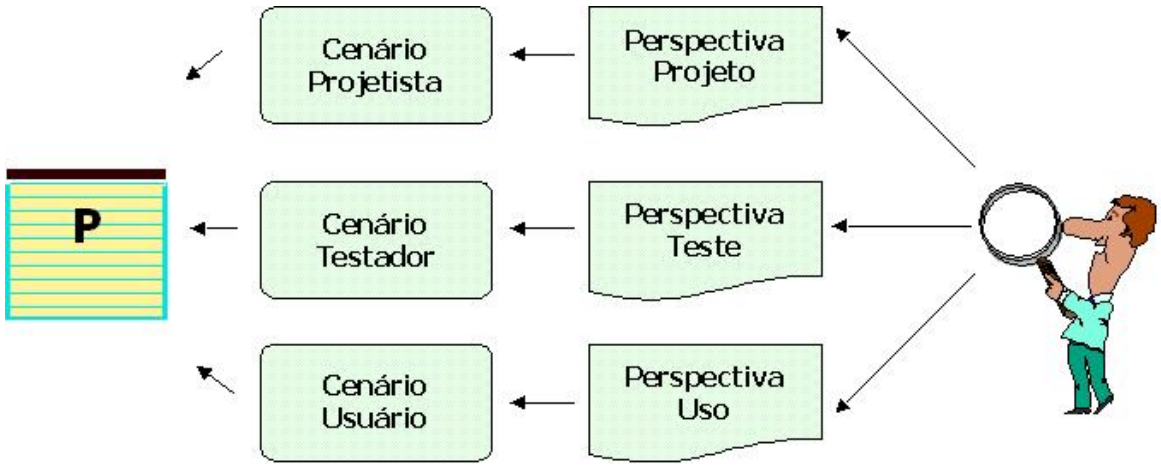
Leitura Baseada em Perspectiva (PBR) II

- Várias leituras podem ser feitas no Documento de Requisitos.
  - O projetista que usa o DR para gerar o projeto do sistema.
  - O testador que, com base no DR, deve gerar casos de teste para testar o sistema quando este estiver implementado.
  - O usuário que verifica se o DR está capturando toda funcionalidade que ele deseja para o sistema.



Leitura Baseada em Perspectiva (PBR) III

- Cada revisor possui um cenário para guiar seu trabalho de revisão.
- Todo cenário consiste de duas partes:
  - Construir um modelo do documento que está sob revisão a fim de aumentar o entendimento sobre o mesmo.
  - Responder questões sobre o modelo, tendo como foco itens e problemas de interesse da organização.



PBR – Visão do Usuário

- Definir um conjunto de funções que o usuário esteja apto a executar.
- Definir o conjunto de entradas necessárias para executar cada função e o conjunto de saídas que são geradas pelas funções.
  - Isso pode ser feito escrevendo todos os cenários operacionais que o sistema deve executar.
- Iniciar com os cenários mais óbvios até chegar nos menos comuns ou condições especiais.
- Ao fazer isso, faça a você mesmo as perguntas a seguir.
  Sugestão: Usar como modelo *Casos de Uso*.

PBR – Visão do Usuário: Questões

- Todas as funções necessárias para escrever os cenários estão especificadas no documento de requisitos ou na especificação funcional?
- As condições para inicializar os cenários estão claras e corretas?
- As interfaces entre as funções estão bem definidas e compatíveis (por ex., as entradas de uma função têm ligação com as saídas da função anterior)?
- Você consegue chegar num estado do sistema que deve ser evitado (por ex., por razões de segurança)?
- Os cenários podem fornecer diferentes respostas dependendo de como a especificação é interpretada?
- A especificação funcional faz sentido de acordo com o que você conhece sobre essa aplicação ou sobre o que foi especificado em uma descrição geral?

PBR – Visão do Testador

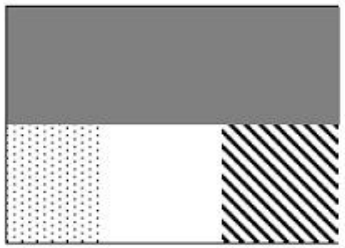
- Para cada especificação funcional ou requisito gere um conjunto de casos de teste que assegure que a implementação do sistema satisfaz a especificação funcional ou o requisito.
- Use sua abordagem de teste normal e critérios de teste.
- Ao fazer isso, faça a você mesmo as perguntas a seguir para cada teste.
  Sugestão: Usar como critérios de teste *Particionamento de Equivalência*, *Análise do Valor Limite*.

PBR – Visão do Testador: Questões

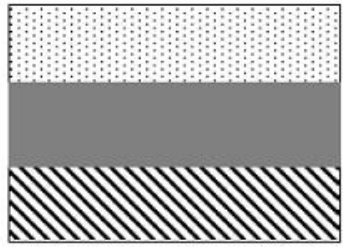
- Você tem toda informação necessária para identificar o item a ser testado e o critério de teste?
- Você pode gerar um bom caso de teste para cada item, baseando-se no critério?
- Você tem certeza de que os testes gerados fornecerão os valores corretos nas unidades corretas?
- Existe uma outra interpretação dos requisitos de forma que o programador possa estar se baseando nela?
- Existe um outro requisito para o qual você poderia gerar um caso de teste similar, mas que poderia levar a um resultado contraditório?
- A especificação funcional ou de requisitos faz sentido de acordo com aquilo que você conhece sobre a aplicação ou a partir daquilo que está descrito na especificação geral?

Vantagens PBR

- Cada revisor vai ler o Documento de Requisitos com olhos diferentes.
- Benefícios:
  - Determina uma responsabilidade específica para cada revisor.
  - Possibilita que o revisor melhore sua experiência em diferentes aspectos do documento de requisitos.
  - Melhora a cobertura de defeitos.
  - Assegura que perspectivas importantes sejam contempladas.



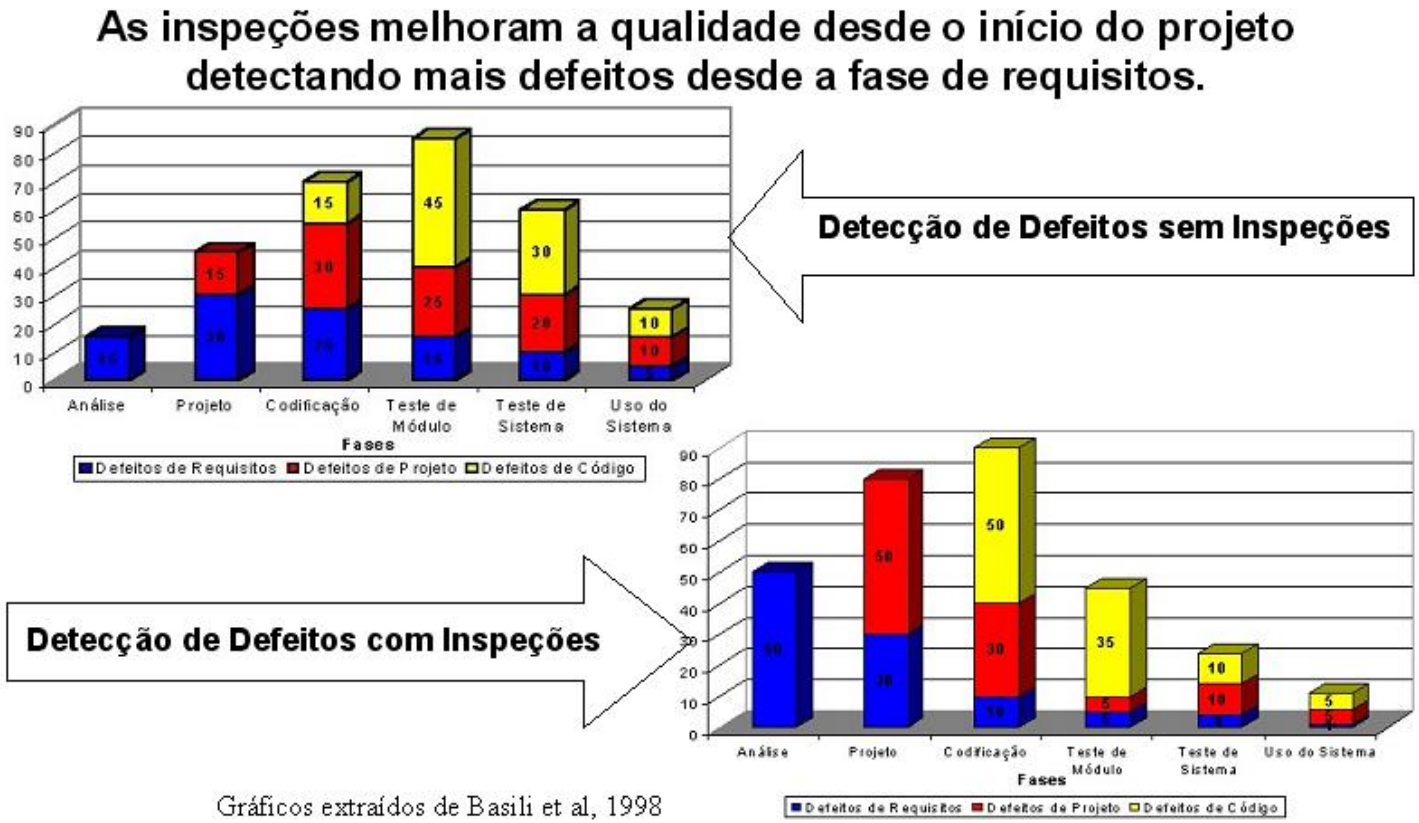
cobertura ad-hoc



cobertura baseada em perspectiva

Vantagens - Inspeção em DR

- Detecção antecipada de defeitos.



Gráficos extraídos de Basili et al, 1998

Vantagens - Inspeção em DR

- Produtividade e diminuição do custo.

Inspeções melhoram produtividade por acharem defeitos quando eles são mais baratos para corrigir.

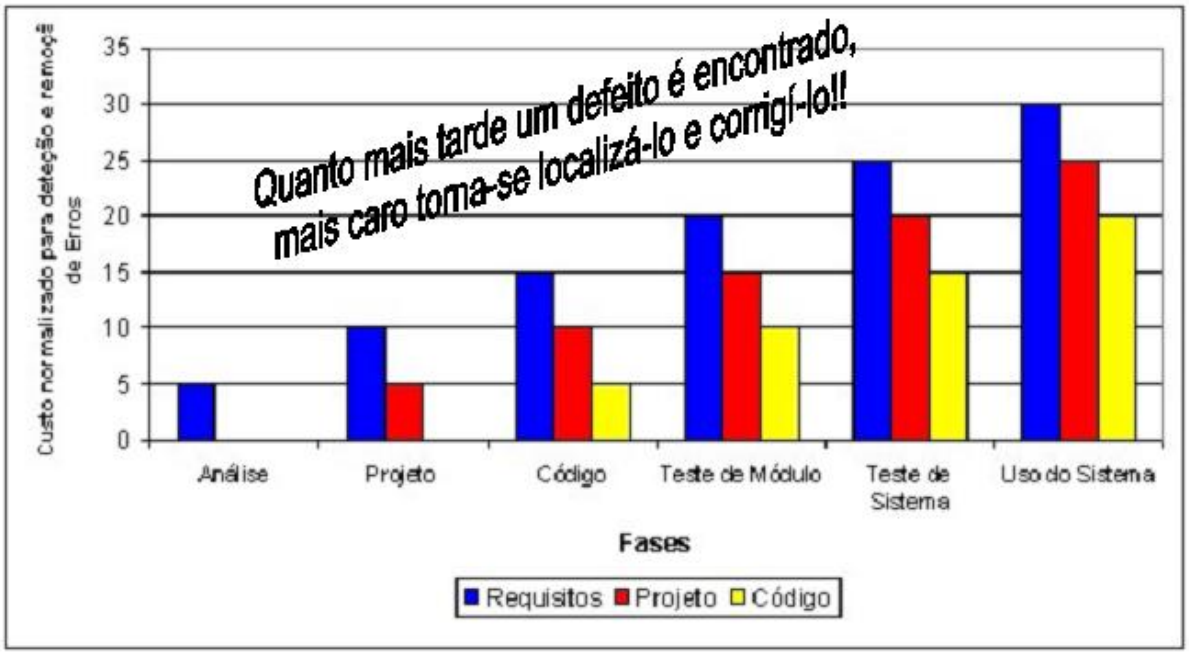


Gráfico extraído de Basili et al, 1998

Concluindo... Inspeção...

- Inspeção não requer necessariamente a execução do sistema e assim pode ser usada antes da implementação estar concluída.
  - Pode ser aplicada em qualquer representação do sistema (requisitos, projeto, dados de testes, ...)
- Quando um erro é encontrado durante a inspeção, é conhecida também sua natureza e localização.
  - Isso não ocorre com Teste de Software.

Concluindo... Inspeção...

- Inspeção e Teste são técnicas de V&V complementares. Ambas devem ser usadas!!!
  - Inspeções podem checar a conformidade com especificação mas não a conformidade com os requisitos reais do cliente!
  - Inspeções não checam características não funcionais como desempenho, usabilidade...

Exercício de Fixação