

- Meio efetivo para melhorar a **qualidade** de software.
 - **Filtro** para o processo de Engenharia de Software.
- Podem ser aplicadas em vários **pontos** durante o desenvolvimento do software.
- Maneira de usar a **diversidade** de um **grupo de pessoas** para:
 - Apontar melhorias necessárias ao produto.
 - Confirmar as partes de um produto em que uma melhoria não é desejada ou não é necessária.
 - Realizar um trabalho técnico de qualidade mais uniforme de forma a torná-lo mais administrável.

- Encontrar erros durante o processo de desenvolvimento, de forma que eles não se transformem em defeitos depois da entrega do software.
- Descoberta **precoce** dos erros.
 - Melhoria da qualidade já nas primeiras fases do processo de desenvolvimento.
 - Aumento da produtividade e diminuição dos custos.
 - Erros são detectados quando sua correção é mais barata.



- É uma oportunidade de treinamento.
 - **Aprender por experiência.**
 - Participantes aprendem as razões e padrões em descobrir erros.
 - Participantes aprendem bons padrões de desenvolvimento de software.
- Com o decorrer do tempo....
 - A revisão auxilia os participantes a desenvolver produtos mais fáceis de entender e de manter.

- **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.



- Métodos de RTF:

- Inspeção
- Walkthrough
- Peer-Review



- 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 Documentos de Requisitos.**
- **Inspeção em Código-fonte.**

Como conduzir uma inspeção?



- Detecção antecipada de defeitos (inspeção de requisitos).
- Aprende-se pela experiência.
 - Participantes aprendem os padrões e o raciocínio utilizado na detecção de defeitos.
 - Participantes aprendem bons padrões de desenvolvimento.
- A longo prazo...
 - A inspeção convence os participantes a desenvolverem produtos mais compreensíveis e mais fáceis de manter.

As inspeções ajudam a integrar o processo de **prevenção** de defeitos com o processo de **detecção** de defeitos.

- Como detectar defeitos?
 - Lendo o documento.
 - Entendendo o que o documento descreve.
 - Verificando as propriedades de qualidade requeridas.
- Problema:
 - **Em geral não se sabe como fazer a leitura de um documento!!!**
- Razão:
 - Em geral, os desenvolvedores aprendem a escrever documento de requisitos, código, projeto, mas não aprendem a fazer uma leitura adequada dos mesmos.

- Solução:
 - Fornecer **técnicas de leitura** bem definidas.
- Benefícios:
 - Aumenta a relação **custo/benefício** das inspeções.
 - Fornece **modelos** para escrever documentos com maior qualidade.
 - Reduz a **subjetividade** nos resultados da inspeção.

Como conduzir uma inspeção?



- O que é uma técnica de leitura?
 - Conjunto de instruções fornecido ao revisor dizendo **como ler** e **o quê procurar** no produto de software.
- Algumas técnicas de leitura:
 - **Ad-hoc**
 - **Checklist**
 - **Leitura Baseada em Perspectiva (PBR)**
 - **Stepwise Abstraction**



Técnica de Leitura Baseada em Perspectiva (PBR) I

SCE 5764 – Engenharia de Software

Casos de Falhas em Sistemas

Qualidade de Software

Atividades de V&V

Revisões de Software

Revisões Técnicas Formais

Inspeção de Software

Walkthrough

Peer-Review

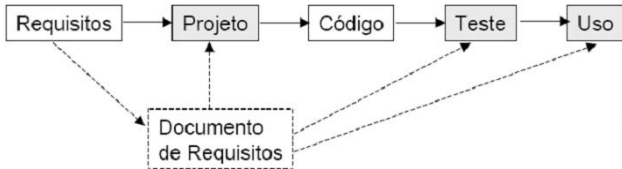
Reunião de Revisão Técnica

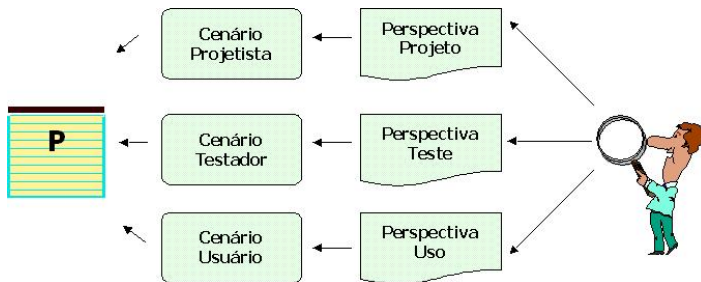
Teste de Software

Conclusão

- Técnica de leitura proposta para detectar defeitos em especificações de requisitos.
- Faz com que cada revisor se torne responsável por uma **perspectiva** em particular.
- Possibilita que o revisor melhore sua experiência em diferentes aspectos do documento de requisitos.
- Assegura que perspectivas importantes sejam contempladas.
- Cada revisor lê o Documento de Requisitos com **olhos diferentes**.
- Benefícios:
 - Determina uma responsabilidade específica para cada revisor.
 - Melhora a cobertura de defeitos.

- 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.



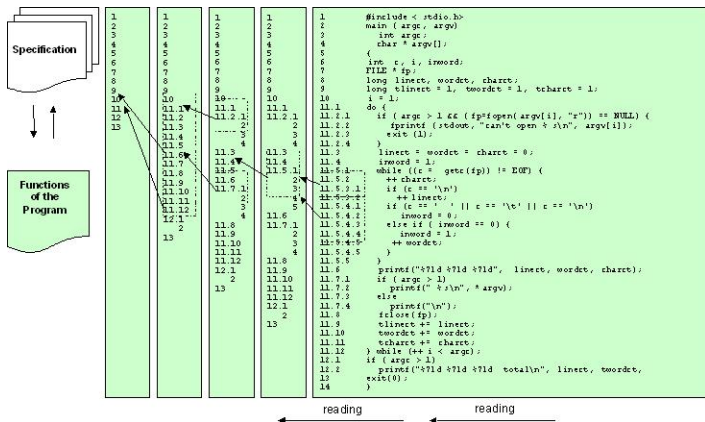


- Técnica de leitura utilizada para detectar defeitos em código-fonte.
 - Code-Reading

- Consiste em desenvolver **abstrações funcionais** a partir do código-fonte, para determinar a funcionalidade do programa.



- Os revisores identificam subprogramas no código-fonte e escrevem suas próprias especificações para os subprogramas (**abstrações de funcionalidade**).
- As **abstrações** construídas devem ser combinadas em uma abstração mais geral, até que se tenha capturado a função completa do programa.
- Inconsistências são detectadas comparando a especificação original com a especificação construída por meio das abstrações.



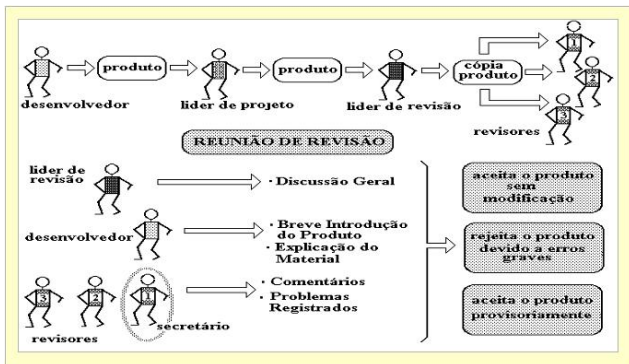
- Procedimento similar ao procedimento para condução de uma inspeção.
- A diferença fundamental está na maneira como a sessão de revisão é conduzida.
 - Em vez de ler o programa ou checar os erros por meio de um *checklist*, os participantes **simulam** sua execução.
 - Papel adicional: testador.
 - Elaborar um pequeno conjunto de casos de teste (em papel).
 - Monitorar e controlar os resultados obtidos.



- Conduzida por **pares de programadores**.
 - Mesmo nível de conhecimento.
- Aplicada ao **código**.
- Reuniões com duração de 1 a 2 horas.
 - Somente um programa ou parte dele (rotinas) deve ser revisado.
- Resultados são publicados em um relatório informal.
 - Não faz parte da documentação oficial do projeto.



- Independentemente do formato da RTF, toda reunião de revisão deve seguir as seguintes recomendações:
 - Envolver de 3 a 5 pessoas.
 - Deve haver uma preparação para a reunião.
 - A preparação não deve exigir mais de 2 horas de trabalho de cada pessoa.
 - A reunião deve durar menos de 2 horas.
 - Deve-se focalizar uma parte **específica** do software.
 - Maior probabilidade de descobrir erros.



- Revise o **produto**, não o produtor.
- Fixe e mantenha uma **agenda**.
- **Limite** o debate e a refutação.
- Relacione as **áreas problemáticas**.
- Faça **anotações** por escrito.
- Limite o número de participantes e insista em uma **preparação antecipada**.
- Desenvolva uma lista de conferência (**checklist**) para cada produto que provavelmente será revisado.
- Atribua **recursos** e uma **programação de tempo** para as revisões.
- Realize um **treinamento** significativo para todos os revisores.
- Reveja suas antigas revisões.