

.....

Aula 3 – Verificação e Validação

Prof. Paulo R. Nietto

.....



Universidade
Anhembi Morumbi

LAUREATE INTERNATIONAL UNIVERSITIES

Objetivos

- Compreender diferenças entre verificação e validação de software
- Importância do processo de verificação e validação
- Exercícios

Processo de Verificação e Validação

- Processo composto de atividades que asseguram que o software cumpra com suas especificações e atenda as necessidades dos usuários

Processo de Verificação e Validação

- Processo que realiza:
 - Revisão de requisitos
 - Revisão de projetos
 - Inspeções de código
 - Testes de software
- V&V deve ocorrer em todas as etapas do ciclo de vida do software

Processo de Verificação e Validação

Verificação

Validação

O software foi construído corretamente?

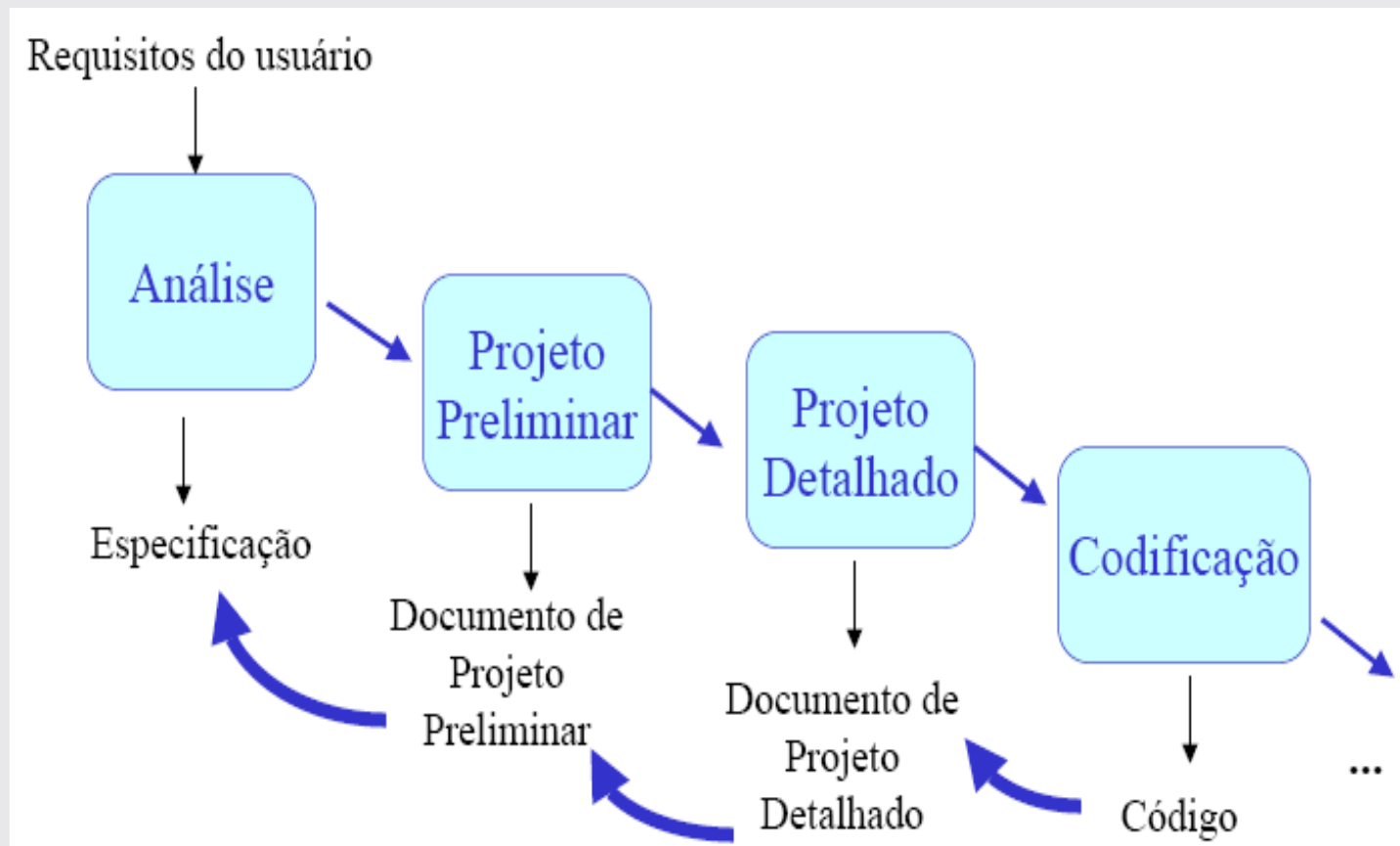


Construímos o software correto?

Verificação

- Checar se o software cumpre suas especificações:
 - Requisitos funcionais
 - Requisitos não funcionais

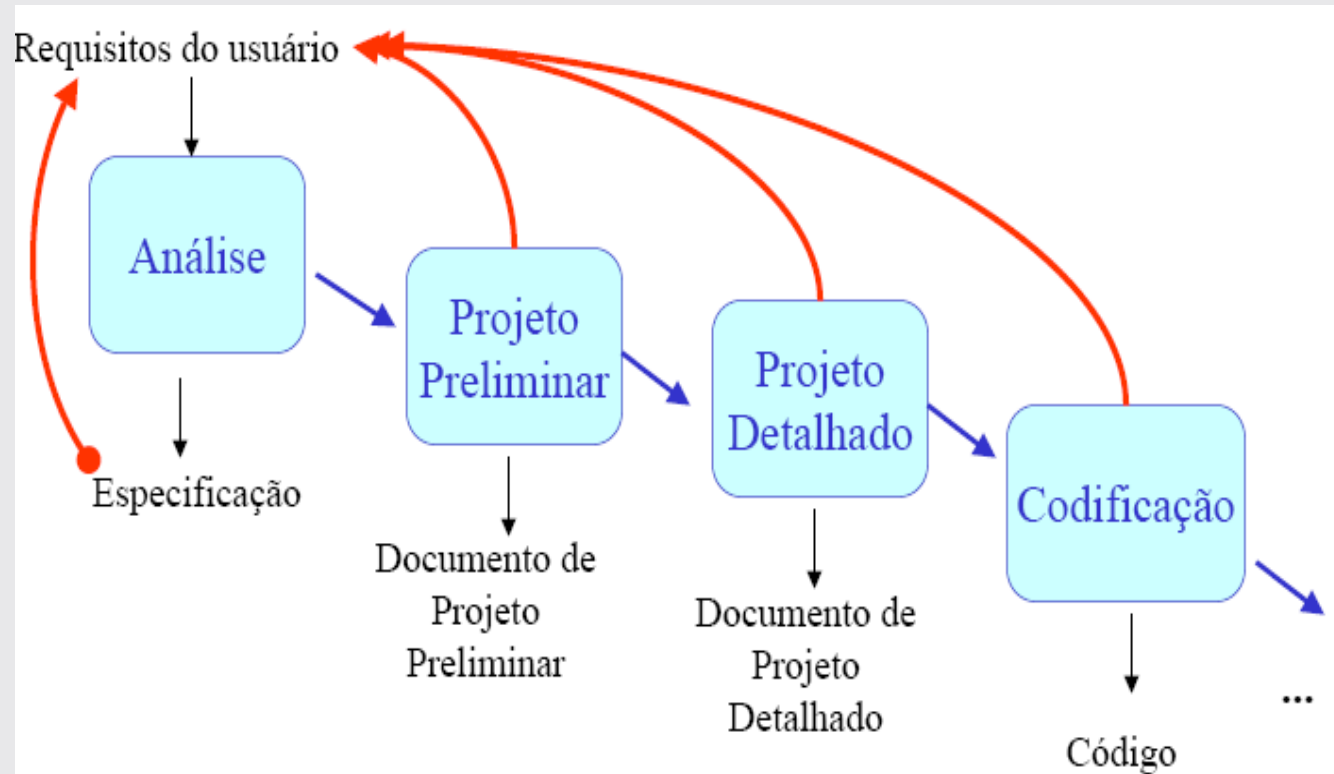
Verificação



Validação

- Assegurar que o software atenda as expectativas do cliente
- Garantir que o software faz o que o cliente espera que faça, exatamente como foi solicitado

Validação



Verificação & Validação

- Análise estática
 - Não requerem a execução propriamente dita do produto.
 - Podem ser aplicadas em qualquer produto intermediário do processo de desenvolvimento.
 - Documento de requisitos, diagramas de projeto, código-fonte, planos de teste, ...
 - As revisões são o exemplo mais clássico de análise estática.

Verificação & Validação

- Análise dinâmica
 - Requerem a execução do produto:
 - Código ou representações executáveis do sistema.
 - As atividades de simulação e teste constituem uma análise dinâmica do produto.

Técnicas de V&V

- Inspeções de software
- Testes de software

Inspeções de software

- Inspeções são técnicas estáticas de V&V realizadas sem que seja preciso executar o software
- Aplicadas em:
 - documento de requisitos
 - diagramas de projeto
 - código fonte – apenas correspondência com a especificação
- Podem ser realizadas em todas as etapas do ciclo de vida de desenvolvimento do software

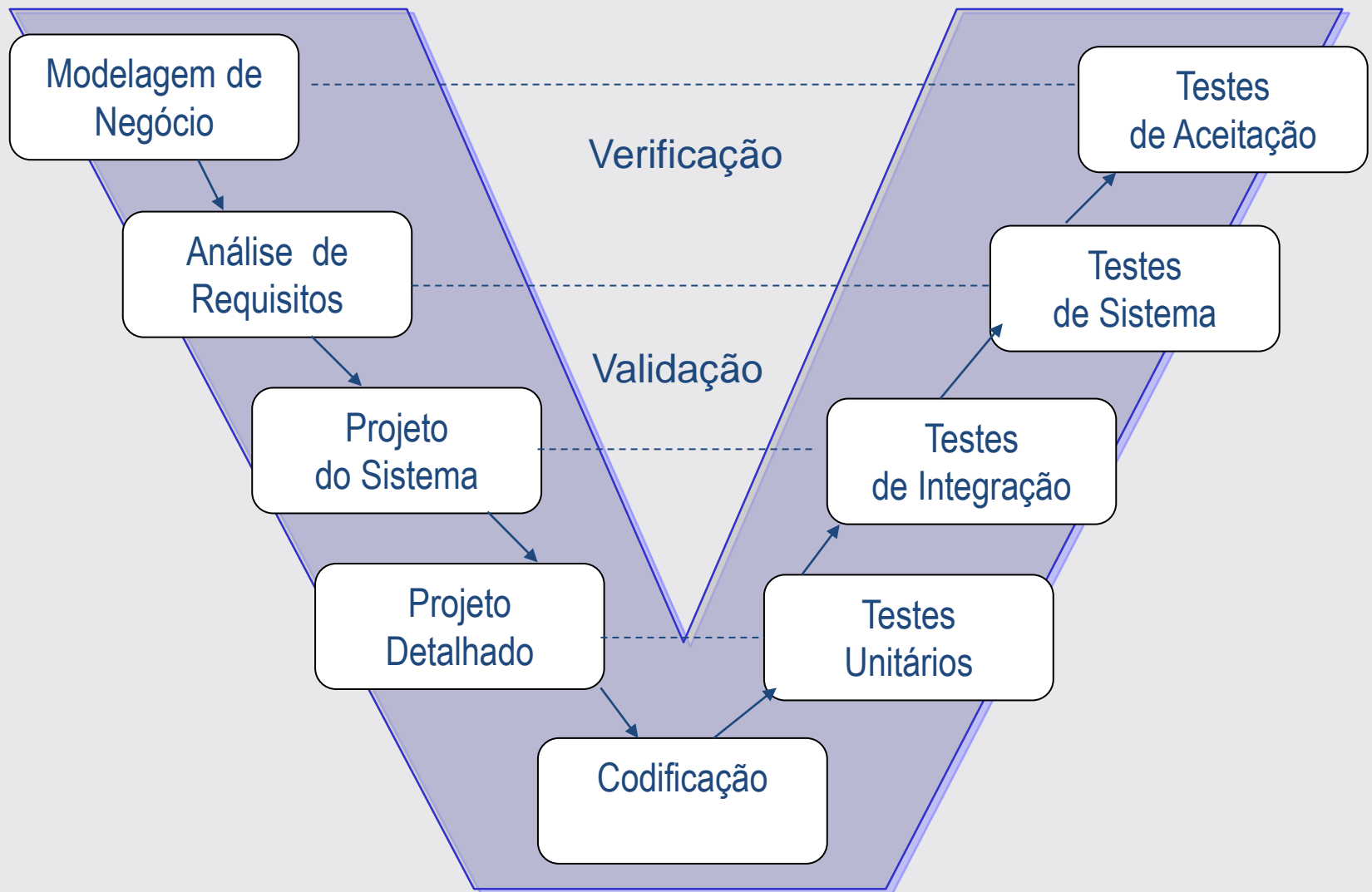
Testes de software

- Testes são técnicas dinâmicas de V&V aplicáveis somente quando o software esta sendo executado
- Executar o software com dados examinar o resultados e comportamento do software
 - Funciona como esperado?
- Podem ser executados somente quando existirem protótipos ou versões do software

Processo de V&V X Ciclo de Vida

- Inspeções: podem ser usadas em todas as fases do ciclo de vida
- Testes: somente quando existir um protótipo ou versão do executável
- Um tipo de técnica não inviabiliza a outra!
 - Devem ser usadas em conjunto!!!

Modelo em “V”



Processo de Verificação e Validação

- Meta:
 - Comprovar que o sistema de software é adequado a seu propósito
- Será possível gerar um software totalmente livre de erros???
 - Não! Suficientemente bom para o uso pretendido.

Planejamento de V&V

- Processo V&V é dispendioso:
 - Sistemas de tempo real, por exemplo, consomem 50% orçamento gasto em V&V
- Necessário planejamento cuidadoso para obter **bons resultados dentro do orçamento**

Técnicas de teste estático

- Revisão
- Inspeção
- Análise automatizadas

Revisão

- Uma revisão é uma **leitura crítica** do **artefato** e de sua **documentação**, anotando:
 - os problemas encontrados
 - dúvidas e dificuldades de compreensão
 - possibilidades de melhorias

Exemplos de artefatos

- especificações
- modelos
- projetos
- código
- casos de teste
- processos de desenvolvimento
- planos
- padrões
- arquivos de auxílio para o usuário
- documentação para o usuário
- teses, dissertações . .

Revisão

- Atividade na qual **um artefato de software** pode ser **observado** pela equipe do projeto, gerentes, usuários, clientes ou outras partes interessadas.

Objetivo da Revisão

- Garantir que o artefato de software esteja em conformidade com suas especificações
- Verificar se seu desenvolvimento esta de acordo com os planos, padrões e diretrizes aplicáveis para o projeto

Por que fazer Revisões?

- Segundo vários autores a revisão quando é praticada encontra **de 60% a 80% do total dos problemas encontrados**
- Yourdon diz que se **economiza aproximadamente 40% do custo total de desenvolvimento** quando se pratica revisões

Formas para realizar uma Revisão

- Revisão pelo próprio autor(*desktop checking*)
- Revisão por colega (peer review)
- Revisões Progressiva

Revisão pelo próprio autor(*desktop checking*)

- O autor lê e anota todos os problemas encontrados para, **depois**, removê-los
- Inconvenientes:
 - o autor tende a “ler o que acha que está escrito e não o que está de fato escrito”
 - erros de entendimento não são observáveis
 - sujeito à síndrome da “idéia fixa”

Revisão por colega (peer review)

- Um dos colegas do autor **lê e anota** todos os problemas encontrados
 - formulário para anotações
 - ninguém deve ficar ofendido com as anotações
 - o que está em jogo é a qualidade do artefato!

Revisões progressivas

- Seleciona-se um conjunto de colegas que farão a revisão
 - cada um com uma determinada especialidade
 - cada colega lê e anota todos os problemas encontrados dentro de sua especialidade
 - a seguir passa a diante para o próximo colega da lista

Realização de Revisões - Prós

- Simplicidade
- Eficácia/Eficiência
 - apesar de informais, revisões tendem a encontrar um número significativo de problemas
 - quando feitas por pessoas treinadas em aspectos formais (prova da corretude) podem ser muito mais eficazes
 - em uma única revisão identifica-se uma quantidade grande de problemas
- Baixo custo
 - algumas coisas podem ser automatizadas

Realização das Revisões - Contras

- Informalidade
 - a qualidade da revisão depende excessivamente dos revisores (proficiência, cultura?)
- Dificuldade em determinar a confiabilidade da revisão
 - qual é a percentagem dos problemas existentes que tendem a ser encontrados?
 - qual a gravidade dos problemas encontrados?
- Confiabilidade depende do rigor adotado pelo revisor
 - frequentemente não é repetível (ambíguo)
 - a falta de treinamento dos revisores amplifica os problemas decorrentes da informalidade

Inspeção

- Conceito introduzido pela IBM na década de 70
- Uma técnica formal para **revisão visual** de artefatos de software para detectar erros, violações de padrões de desenvolvimento e outros problemas.
- A inspeção é muito rigorosa e os inspetores são treinados em técnicas de inspeção
- Aplicada amplamente para **revisar código de programa**

Objetivos das inspeções

- Verificar se o artefato de software satisfaz as especificações e está de acordo com os padrões aplicáveis
- Identificar desvios
- Coletar dados de engenharia de software como defeito e esforço

Por que fazer Inspeções?

- 67% do total de defeitos durante o desenvolvimento são detectados na Inspeção
- 82% de todos os defeitos achados durante a inspeção referem-se a design e código

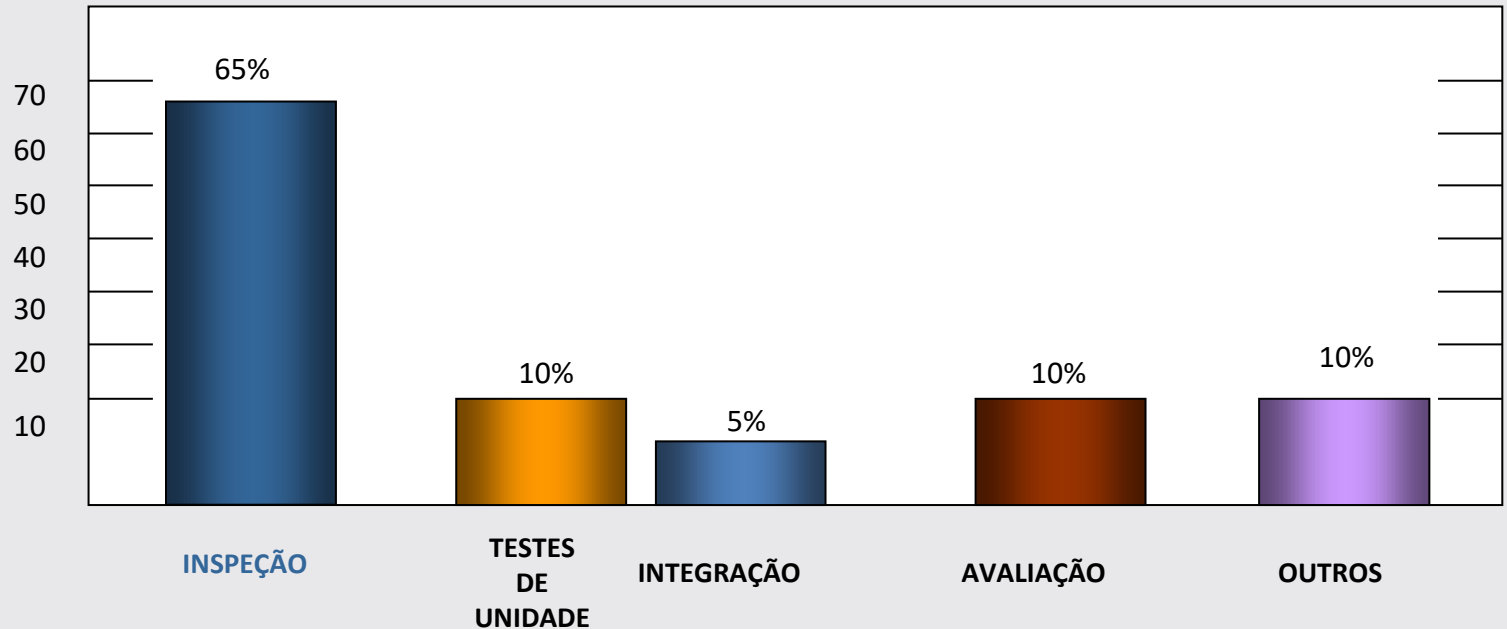
Inspeção de Programa

- Processo formal realizado por uma equipe a partir de um checklist de defeitos pré-definido.
- Objetivo : inspecionar cada linha de código a procura de defeitos ou não cumprimento de padrões

Realização de Inspeções

- Prós
 - processo formal realizado sistematicamente com resultados confiáveis
 - de 60% a 90% dos erros de um programa podem ser encontrados em um única revisão
- Contras
 - Resistência ao uso – pessoal especializado aumenta o custo do projeto

Importância das inspeções na detecção de defeitos



Analísadores automáticos de programas

- São ferramentas que processam o código fonte do programa e chamam a atenção do verificador para anomalias, como seções não utilizadas e variáveis não inicializadas/utilizadas
- Amplamente utilizadas pelas equipes de inspeção
- Exemplo:
 - LINT para UNIX
 - Findbugs (java)
 - Cpplint (google c++)

Ferramentas para automatização de inspeções

- GRIP (GRoupware supported Inspection Process)
- IBIS (Internet Based Inspection System)
- ISPIS (Infra-estrutura de Suporte ao Processo de Inspeção de Software).
 - Esta última sendo tecnologia nacional desenvolvida na COPPE/UFRJ,

Análise automatizada - Vantagens

- Equipe pode se focar mais na verificação manual da funcionalidade / algoritmo
- O uso da ferramenta resulta em completude e consistência
- Economiza esforço
- Encontra defeitos que não são possíveis de serem encontrados manualmente

Testes estáticos X Dinâmicos

- Vantagens dos testes estáticos sobre os dinâmicos
 - Detecção de defeitos o mais rápido possível
 - experiência dos revisores
 - Testes estáticos expõe os defeitos os enquanto os dinâmicos mostram sintomas
 - resultado de um defeito ou efeito colateral de um defeito já conhecido?
 - Teste estáticos encontram mais de um defeito por vez enquanto os dinâmicos um a um
 - ex: programa interrompido

Testes estáticos X Dinâmicos

- Alguns defeitos podem ser achados somente através do Teste Estático
 - Redundância de Código (quando a lógica não é afetada)
 - Código morto (nunca executado)
 - Violação de padrões de código

Revisão e Inspeção

Item	Revisão	Inspeção
Objetivos	<ol style="list-style-type: none">1. Avalia a conformidade com as especificações2. Garante a integridade	<ol style="list-style-type: none">1. Detectar e identificar defeitos2. Documentação dos defeitos3. Verificar resolução
Qualificação dos participantes	<ol style="list-style-type: none">1. Experts técnicos2. Membros da equipe	<ol style="list-style-type: none">1. Inspetores formalmente treinados
Poder decisório	A equipe de Revisão pede à liderança da equipe de Projeto ou Gerência para levar as recomendações adiante	A equipe declara informa o resultado (Aceite/ Re-trabalho/Verificação/ Re-inspeção)

Saiba Mais

- Leia o artigo sobre inspeção de software:
<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-inspecao-de-software/8037>

Exercício 1

- () Testes devem ser realizados para mostrar a inexistência de defeitos
- () O processo de verificação e validação deve ser independente do processo de desenvolvimento, porém integrado
- () Verificação e validação são técnicas de teste
- () A equipe de teste pode ser formada por profissionais menos experientes e qualificados

Exercício 2

- Explique as diferenças entre verificação e validação. Por que a validação é um processo particularmente difícil.

Exercício 4

- Um gerente da empresa decide usar os relatórios de inspeção de programa como fonte de informação para avaliar seus funcionários que trabalham em desenvolvimento de software. Os relatórios mostram quem cometeu o erro e quem os encontrou durante a inspeção. Esse comportamento pode ser considerado ético? Séria ético que a equipe de inspeção fosse informada previamente que isto aconteceria? Que diferença isso poderia fazer para um processo de inspeção de programa?

Exercício 5

- É comum durante o desenvolvimento de sistemas, realizar os testes até que o orçamento destinado aos testes termine. Então, o sistema é entregue aos clientes. Discuta a ética desta abordagem.

Exercício 6 – Faça uma revisão do texto abaixo

- Um sistema para emissão automática de passagens de trem deve ser construído. Os usuários selecionam o destino desejado e fornecem um número de cartão de crédito. O bilhete é emitido e seu preço é descontado no cartão de crédito fornecido. Quando o usuário aperta o botão «início» é mostrado um menu com vários destinos possíveis e mensagem solicitando ao usuário que escolha um destino. Depois que o destino foi escolhido, é solicitado ao usuário que entre com o cartão de crédito. A validade do cartão é verificada e em seguida é solicitado ao usuário um número de identidade (CPF ou RG). O bilhete é emitido após a transação de crédito ter sido validada.

Referências

- SOMMERVILLE, Ian. Engenharia de Software. São Paulo: Addison-Wesley, 2004
- PRESSMAN, Roger S. Engenharia de Software. São Paulo: Makron Books, 1995.
- CRAIG, R.D. Systematic Software Testing. New York: Artech House, 2002.
- Jeff Tian. Software Quality Engineering - Testing, Quality Assurance, and Quantifiable Improvement. IEEE Computer Society. John Wiley & Sons, Inc. 2005.

.....

Obrigado

.....

paulo.nietto@animaeducacao.com.br



**Universidade
Anhembi Morumbi**
LAUREATE INTERNATIONAL UNIVERSITIES