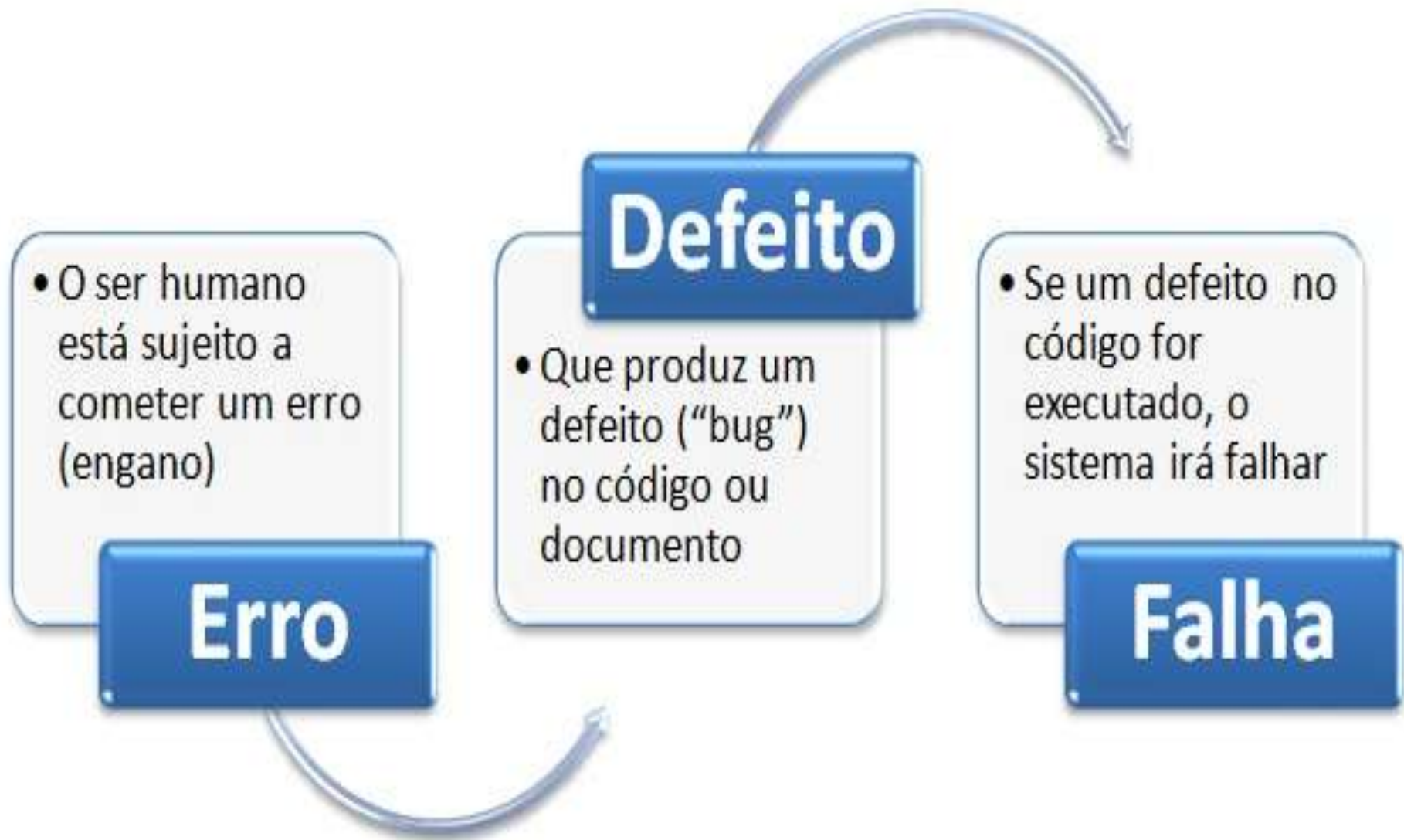


# Aula 3 – Verificação e Validação

## Parte I

Trabalhos de qualidade não podem ser feitos sem concentração, auto-sacrifício, esforço, padronização e as vezes uma certa dúvida do novo.

# Defeito, Erro, Falha?



# Defeito, Erro, Falha?

- Maior parte de Origem humana
- Oriundos da “comunicação”
  - Transporte de informações
  - Transformação de informações
- Podem permanecer nos produtos liberados
- Encontrado geralmente em partes do código raramente executadas
  - Particularidades não testadas ou analisadas
- Percepção de defeito x Custo
  - Início do projeto
    - Custo menor
    - Probabilidade de corrigir corretamente sem causar danos
  - Projeto em andamento
    - Custo aumenta
    - Probabilidade de correção antes de danos menor

# Tipos de defeito

- Tipos de defeito
  - Inicialização
    - Variável ou estrutura de dados inicializada incorretamente
    - Não inicializada
  - Computação
    - Falta de uma expressão
    - Expressão incorreta
  - Controle
    - Erro de fluxo de controle (if – then – else)
  - Interface
    - Defeitos entre módulos e suas utilizações em conjunto
  - Dados
    - Uso incorreto de estrutura de dados
  - Aparência
    - Não causam execução errônea
    - Ex: erro de ortografia em mensagem de saída

# Taxonomia de Defeitos

- Omissão
  - As informações necessárias sobre o sistema foram omitidas
- Fato incorreto
  - Informações de um artefato contradizem especificação, domínio do problema.
- Inconsistência
  - Informações de um artefato contradizem com outros artefatos
- Ambigüidade
  - Possibilidade de interpretar informações de diferentes maneiras
- Informação estranha
  - Informações são fornecidas
    - Não são necessárias
    - Úteis
    - utilizáveis

# Garantia de Qualidade de Software

Conjunto de atividades aplicadas durante o processo de desenvolvimento

## Objetivos

- garantir que processo e produto atinjam níveis de qualidade pré-estabelecidos
- garantir a conformidade com requisitos, padrões

# Garantia de Qualidade de Software versus defeitos

**De que forma  
solucionar ?**

solucionar ?  
De que forma

# Verificação, validação e teste?

**São conceitos  
diferentes?  
Defina?**

Defina;  
diferentes;  
São conceitos



# Verificação, validação e teste?

- **Validação**
  - Assegura que o produto final corresponda aos requisitos do software
    - “Estamos construindo o produto certo?”
    - “Atende as expectativas do cliente?”
- **Verificação**
  - Assegura que o produto possui consistência, completitude e corretitude em cada fase e entre as fases consecutivas do ciclo de vida
    - “Estamos construindo corretamente o produto?”
    - “Está de acordo com as especificações?”
      - Atende os requisitos funcionais e não funcionais especificados?
- **Teste**
  - Examina o comportamento do produto através de sua execução
    - Funcional ou não funcional

*A pressão leva o desenvolvedor de software a consertar um erro e, ao mesmo tempo, introduzir mais dois”*

Pressman

# Verificação e Validação

- Ter certeza de que o sistema está adequado a seu propósito
  - Bom o suficiente para o uso pretendido
- Confiabilidade x propósitos
  - Função do software
    - Utilização na organização (quão crítico é seu uso)
      - Controle de equipamento hospitalar
      - Fabricação de papel
  - Expectativas do usuário
  - Ambiente de mercado
    - Existe concorrência?
    - Qual a qualidade da concorrência?

# Verificação e Validação – Revisões Técnicas Formais

- Objetivos
  - Descobrir erros lógicos, funcionais
  - Verificar o cumprimento dos requisitos
  - Garantir cumprimento de padrões
  - Garantir um desenvolvimento uniforme
  - Facilitar o gerenciamento dos projetos
  - Facilitar o controle de qualidade
  - Tornar o sistema conhecido por mais pessoas
- Técnica disponível para fases iniciais do desenvolvimento
- Principais métodos: inspeções, walkthrough e PBR (perspective-based reading)

# Verificação e Validação – Revisões Técnicas Formais

- Inspeção
  - modelos são examinados em detalhe por técnicos
    - detecção de erros
    - Violação de padrões de programação e especificação
    - Não cumprimento dos requisitos
  - Ex: checklist
- Walkthrough
  - reunião formal de revisão pré-agendada
  - concentra-se em um módulo ou parte do software
  - onde o produtor “caminha” através do produto explicando o material e os revisores levantam questões
  - Baseada em um conjunto de casos de teste

# Verificação e Validação – Revisões Técnicas Formais

- Perspective-Based Reading
  - encontro formal
    - modelo é apresentado a técnicos e usuários
    - Instruções de como deve-se olhar/ler um produto de software são dadas aos leitores
    - Leitores avaliam o documento em uma perspectiva de revisão
    - Necessidade de mais de uma perspectiva para uma revisão efetiva
    - Resultado da revisão documentado e divulgado
  - Perspectivas de revisão
    - Baseadas no projeto
    - Baseadas no teste
    - Baseadas no uso

# Verificação e Validação – Revisões Técnicas Formais

## – Perspectivas de revisão

- Baseadas no projeto

- Todos os objetos necessários estão definidos?
- A especificação de todos os requisitos estão claras o suficiente ou algo ainda ficou subentendido?
- Interfaces especificadas e consistentes?
- A especificação dos requisitos faz sentido?

- Baseadas no teste

- Todas as informações necessárias para planejar os testes estão informadas e claras o suficiente?
- Existe problemas quanto ao entendimento das especificações? É possível entender e realizar de forma diferente o mesmo requisitos? Isto afetaria o seu teste?

- Baseadas no uso

- Todas as condições de iniciação de um cenário operacional estão claras?
- É possível iniciar um cenário em um estado que deve ser evitado?
- Existe diferentes interpretações para os cenários?

# Verificação e Validação – Revisões Técnicas Formais

- Regras ou diretrizes para o sucesso de revisões técnicas formais
  - Revise o produto e NÃO revise o produtor
  - Mantenha uma agenda (FOCO)
  - Faça anotações
  - Desenvolva um checklist
    - Evidenciar o que já passou por revisão
    - Evidenciar o que falta passar por revisão
  - Atente-se ao número de participantes
  - Atente-se ao tempo gasto pelas revisões
  - Desenvolva um relatório das revisões com os problemas coletados e possíveis soluções para os mesmos

# Exercício I - Checklist

Em duplas:

- 1 – Escolha uma ERS de um componente do grupo. Pode ser do estágio ou engenharia II
- 2 – Troque a ERS com outro grupo
- 3 - Utilize-se da ERS trocada para aplicar o checklist dado realizando uma revisão técnica por intermédio de inspeção da qualidade dos requisitos coletados e documentados
- 4 – Desenvolva o relatório final de resultados
- 5 – Entregue novamente a ERS e resultados para a equipe portadora do documento
- 6 – Análise os resultados do documento original de sua equipe e os problemas encontrados



## Exercício II – Walkthrough

Em duplas:

- 1 – Requisito de Software é apresentado
- 2 – Apresentação
- 3 – Levantamento de questões