



Universidade Federal de Santa Catarina
Graduação em Tecnologias da Informação e Comunicação
Engenharia de Software II
Professora: Tatiana Nilson dos Santos

ENGENHARIA DE SOFTWARE II

Prof^a.: Tatiana Nilson dos Santos
email: tatiana.santos@ufsc.br
Turma: DEC7130 - 04652

VERIFICAÇÃO E VALIDAÇÃO



Processo de verificação e validação

- ✓ Mesmo um sistema que foi minuciosamente projetado e construído não está livre de apresentar erros/falhas;
- ✓ Para isso, antes do sistema ser entregue ao cliente é necessário executar uma série de atividades envolvendo a verificação e validação do mesmo;

Processo de verificação e validação

- ✓ Aplicado em todas as etapas do processo de software;
- ✓ Principais objetivos:
 - Descobrir defeitos no sistema;
 - Aferir se o sistema é útil e usável em operação;
 - Assegurar que o software cumpra com suas especificações e atenda as necessidades do cliente;
 - Criar confiança de que o software é adequado ao propósito inicial;

Processo de verificação e validação

Por que utilizar técnicas de verificação e validação?

- ✓ Resultados de estudos experimentais evidenciam benefícios da utilização destas técnicas no desenvolvimento de software;
- ✓ A utilização destes métodos na indústria tem mostrado resultados positivos considerando tanto produtividade quanto qualidade.

Processo de verificação e validação

- ✓ Inspeções aumentam significativamente a produtividade, qualidade e estabilidade dos projetos [FARGAN, 1976];
- ✓ Uma combinação de diferentes técnicas de VER&VAL apresenta melhor desempenho do que qualquer método isoladamente [HETZEL, 1976 e MEYER, 1978];
- ✓ Qualidade melhora a produtividade [MILLS, 1983];
- ✓ Corrigir um defeito após a entrega do produto é frequentemente 100 vezes mais caro que corrigi-lo durante as atividades de requisitos e projeto do sistema [BOEHM e BASILI, 2001];
- ✓ Testes podem provar a presença de erros, não sua ausência [DIJKSTRA, 1970].

Processo de verificação e validação

- ✓ **Não significa que seja completamente livre de defeitos:**
 - Tem que ser suficientemente bom para a utilização pretendida;
- ✓ **Confiança** depende:
 - Propósito do sistema: quão crítico é o software para a organização;
 - Expectativas do usuário: usuários podem ter expectativas relativamente baixas a certos tipos de software;
 - Ambiente de *marketing*: lançar produto rapidamente no mercado pode ser mais importante que encontrar defeitos no programa.

Processo de verificação e validação

Para que serve esse processo?

- ✓ Descobrir erros de função, de lógica ou implementação;
- ✓ Verificar se o software atende aos requisitos especificados;
- ✓ Garantir que o software foi representado de acordo com os padrões pré-definidos;
- ✓ Garantir que o software seja desenvolvido de maneira uniforme;
- ✓ Desenvolver projetos mais generalistas, permitindo assim as técnicas de reuso de software.

Terminologias

- ✓ **Defeito:** deficiência mecânica ou algorítmica que, se ativada, pode levar a uma falha;

“Faz parte do produto, é algo que está implementada no código de maneira errada.”

- ✓ **Erro:** item de informação ou estado de execução inconsistente;

“Teve um valor diferente do que esperado.”

- ✓ **Falha:** evento notável em que o sistema viola suas especificações;

“Houve um retorno de um valor não esperado, como ‘null’ e por causa disso ocasionou uma falha no sistema.”

Verificação

Fizemos o software corretamente?

- ✓ Avalia se os requisitos e funcionalidades documentadas foram implantados;
- ✓ Assegurar que o projeto está de acordo com os requisitos levantados junto ao cliente;
- ✓ Envolve checar se o software cumpre com suas especificações.

Práticas de V&V

Verificação estática

Inspeção de software

Analisar representação estática do sistema para descobrir problemas

Pode ser complementada por análise de documentação e de código baseada em ferramentas

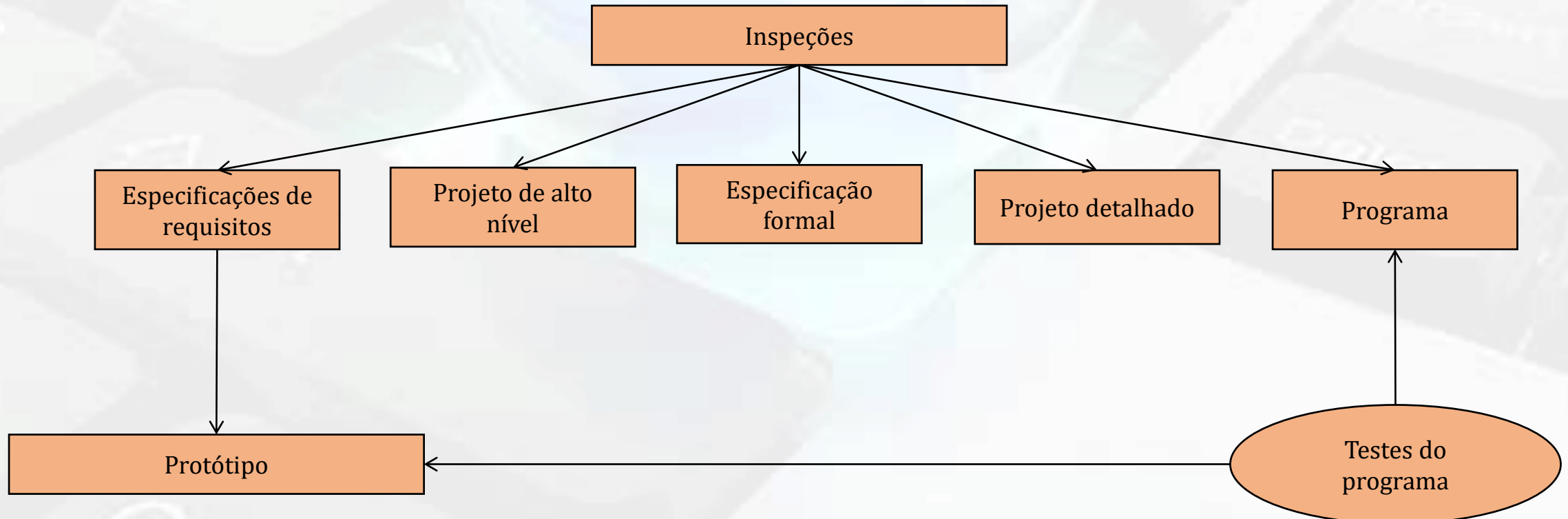
Teste de software

Exercitar o sistema e observar comportamento do produto

Sistema é executado com dados de teste e comportamento operacional observado

Verificação dinâmica

Inspeção de Software



Teste de software

O software tem defeito?

- ✓ Técnica dinâmica de verificação e validação, pois o software é executado com dados de teste;
- ✓ Utiliza-se também alguns *checklists* para complementar o processo de teste;
- ✓ Devem ser usados em conjunto com a verificação estática (inspeção);

Teste de software

✓ Teste de Desenvolvimento

- Os componentes são testados pelas pessoas que os desenvolvem;
- Ferramentas são usadas para reexecutar os testes;

✓ Teste de Sistema

- Os componentes integrados para criar um sistema (ou subsistema) são testados;

✓ Teste de Aceitação

- O sistema é testado com dados do cliente.

Validação

Fizemos o software correto?

- ✓ Avalia se o que foi entregue atende as expectativas e necessidades do cliente, ou seja, se o sistema é realmente o que o cliente pediu e está pagando;
- ✓ Em muitos casos, é o próprio cliente que efetiva o processo de validação final do software;

Validação

- ✓ Problemas que podem ser resolvidos no processo de validação:
 - Não seguimento das normas de qualidade do projeto e da empresa;
 - Descrição pouco clara dos requisitos;
 - Falha na modelagem de requisitos;
 - Conflito entre requisitos que não foram detectados no processo de análise;
 - Requisitos não realistas;
 - Falta de informação.

Validação



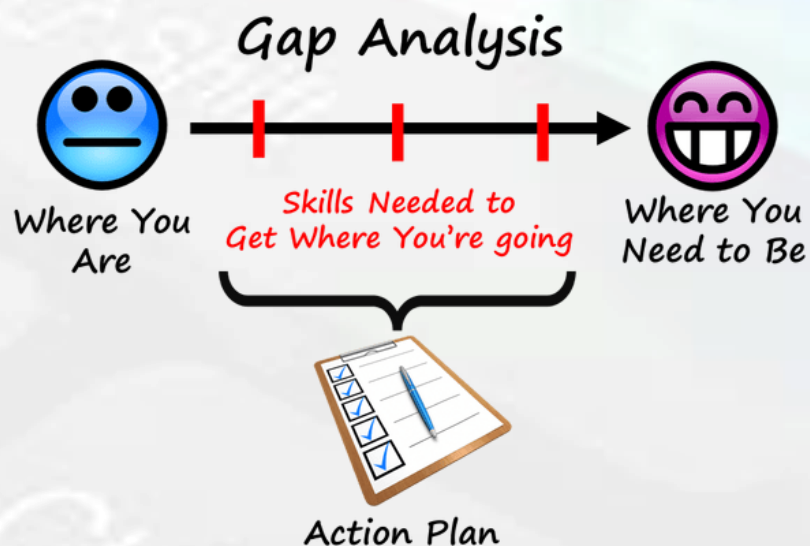
Validação

✓ Técnicas de Validação

- **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

GAP Analysis

- ✓ Método de avaliar as diferenças de desempenho entre sistemas de uma empresa para determinar se os requisitos de negócios estão sendo atendidos e, se não, quais etapas devem ser tomadas para garantir que sejam atendidas com êxito.



Gap refere-se ao espaço entre "onde estamos" (o estado atual) e "onde queremos estar" (o estado alvo).

GAP Analysis

✓ Algumas ferramentas comuns:

- **McKinsey 7S Framework**: usada para determinar aspectos específicos de uma empresa que estão atendendo às expectativas (7S – estratégia, estrutura, sistemas, equipe, estilo, habilidades e valores compartilhados).
- **Matriz SWOT**: usada para identificar os fatores internos e externos que afetam a eficácia e o sucesso de um produto, projeto ou pessoa.

Conclusão

Verificação	Estamos a construir certo o produto?	Software tem de cumprir a especificação
Validação	Estamos a construir o produto certo?	Software tem de fazer o que o usuário quer
Teste	O software possui defeitos?	Comportamento do software é examinado por meio de sua execução
Inspeção	O software possui erros?	Software não pode apresentar informações inconsistentes



Atividade

1. Debater e listar os principais fatores que dificultam, na sua opinião, a realização de verificação e validação:
 - a. Em um software implantado no ambiente organizacional há mais de 5 anos;
 - b. Em um software que acabou de ser lançado no mercado;