

Verificação e Validação

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

Verificação e Validação (V&V)

- Objetivos da verificação e validação
 - Mostrar que o software atende a sua especificação
 - Mostrar que o software atende as necessidades do cliente
- Teste é a principal técnica de V&V
 - Técnicas de inspeção e revisão também são usadas

Verificação

- O objetivo é verificar se o software atende aos requisitos funcionais e não funcionais especificados
- Verificação inclui a realização de testes para encontrar erros
- Pergunta principal
 - *Estamos construindo o produto corretamente?*

Validação

- A inexistência de erros não mostra a *adequação operacional* do sistema
 - Deve ser feita a **validação** com o cliente
- A validação procura assegurar que o sistema atenda as expectativas e necessidades do cliente
- Pergunta principal
 - *Estamos construindo o produto correto?*

Testes de Software

- *“Testes podem somente revelar a presença de defeitos, não a ausência”*
Dijkstra
- Testes fazem parte do processo de verificação e validação (V&V)
 - Devem ser usados em conjunto com a verificação estática (inspeção)

Estágios de Teste

- Teste de Desenvolvimento
 - Os componentes são testados pelas pessoas que os desenvolvem
 - Ferramentas, como o JUnit, são usadas para re-executar os testes
- Teste de Sistema
 - Os componentes integrados para criar um sistema (ou sub-sistema) são testados
- Teste de Aceitação
 - O sistema é testado com dados do cliente

Processo de Teste



- Objetivos das atividades
 - Teste de Desenvolvimento: descobrir erros no início do processo
 - Teste de Sistema: encontrar erros resultantes de interações entre componentes
 - Teste de Aceitação: garantir que o sistema satisfaz as necessidades dos usuários

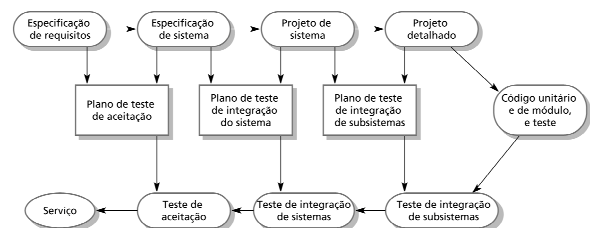
Dos Requisitos ao Teste

- Verificação e validação deve ocorrer durante e depois do desenvolvimento
- O Modelo V ilustra as atividades de testes durante o desenvolvimento
 - Começa na especificação de requisitos
 - Revisões de arquitetura e projeto
 - Inspeções e testes de código

O Modelo V

- Modelo que integra o desenvolvimento aos testes
- É fortemente baseado no Modelo Cascata
- Os planos de testes são derivados das atividades de desenvolvimento

O Modelo V



Níveis de Confiabilidade

- V&V busca estabelecer a confiança de que o software está pronto para ser usado
- O nível de confiabilidade de software depende
 - Do próprio sistema
 - Das expectativas dos usuários
 - Do ambiente de mercado

Finalidade do Sistema

- Sistema Crítico
 - O nível de confiança é muito maior
 - O software deve ser confiável
- Protótipo
 - O nível de confiança é menor
 - O software pode falhar

Expectativa de Usuários

- Usuários podem já estar acostumados com software de baixa qualidade
- Usuários tendem a aceitar falhas quando os benefícios superam as desvantagens
 - A tolerância a falhas dos usuários diminuí a medida que o software amadurece

Ambiente de Mercado

- Sistemas comerciais devem levar em conta os programas concorrentes
- Em um ambiente competitivo
 - O sistema pode ser liberado mais cedo pelo pioneirismo
 - V&V pode não ter sido bem feita
- Quando clientes não querem pagar caro pelo produto
 - Eles geralmente aceitam alguns defeitos

Inspeção de Software

Técnicas de V&V

- Testes de software é a maneira mais comum de se verificar se o programa atende a sua especificação
- Para complementar os testes, é comum o uso de inspeção de software ou revisão por pares

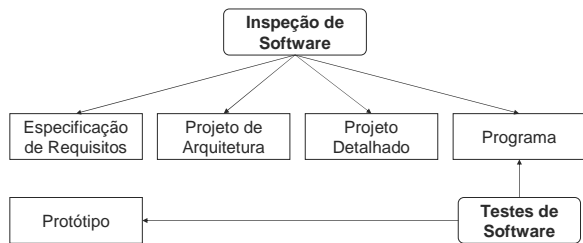
Inspeções e Revisões

- Técnica estática de V&V
 - Não precisa executar o programa
- Pode ser usada em qualquer atividade de desenvolvimento
 - Requisitos, projeto, código fonte, etc.
- Pode ser semi-automatizada por análise estática
 - Análise não automatizada é cara

Testes de Software

- Teste é uma técnica dinâmica de V&V
 - Consiste em executar uma implementação com dados de teste
- Pode ser usada para avaliar os requisitos não funcionais
 - Desempenho, confiabilidade, segurança, etc.

[Verificação Estática e Dinâmica]



[Vantagens da Inspeção]

- Muitos defeitos diferentes podem ser descobertos em uma única inspeção
 - Um teste revela um defeito e oculta vários
- Versões incompletas do sistema podem ser inspecionadas
- Permite encontrar problemas em outros atributos de qualidade do software
 - Uso de um algoritmos mais eficiente, padrão de projeto, etc.

[Limitações de Inspeção]

- Inspeção não é adequada
 - Para descobrir defeitos
 - Para demonstrar se o software é útil
 - Para verificar requisitos não funcionais, como desempenho, segurança, etc.
- Inspeção é uma técnica cara
- Ela não permite validar com o cliente
 - Somente verifica a correspondência entre a especificação e o software

[Bibliografia]

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
 - Seção 2.2.3 Validação de Software
 - Cap. 8 Testes de Software