

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



```
import java.io.*;
import java.net.*;
import java.util.*;

public class Protection {
    public static void main(String[] args) throws IOException {
        OutputStream out = new DataOutputStream(
            DataOutputStream.getOutputStream());
        long q1 = Math.random();
        double protected1 = Protection.mak
        byte[] protected2 = Protection.mak
        long q2 = Math.random();
        double q2 = Math.random();
        byte[] protected2 = Protection
        out.writeUTF(user);
        out.writeInt(protected1.length);
        out.write(protected2);
        out.flush();
    }

    public static void main(String[] args) {
        String host = "127.0.0.1";
        int port = 8080;
        String user = "John";
        String password = "sh";
        Socket s = new Socket(host, port);
        Client client = new Client(s);
        client.sendAuthent
    }
}
```

■ 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 da maneira correta?*



[Validação]

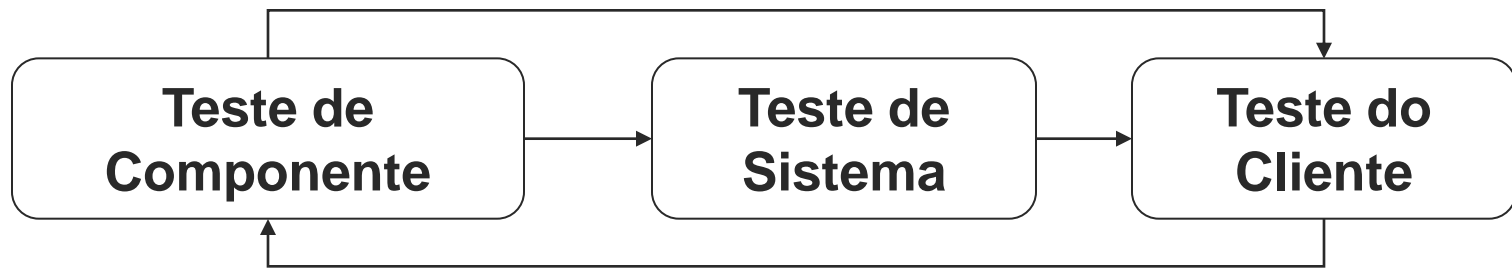
- A inexistência de erros não mostra a *adequação operacional* do sistema
 - Deve ser feita a **validação** com o cliente
- A validação procura assegurar que o sistema atenda as expectativas e necessidades do cliente
- Pergunta principal
 - *Estamos construindo o produto correto?*



[Estágios de Teste]

- Teste de Componente
 - 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 são integrados e o sistema é testado.
- Teste do Cliente
 - O sistema é testado com dados do cliente

Processo de Teste



■ Objetivos das atividades

- Teste de Componente: descobrir erros no início do processo
- Teste de Sistema: encontrar erros pela interações entre componentes
- Teste do Cliente: garantir que o sistema satisfaz as necessidades dos usuários

Desenvolvimento Incremental

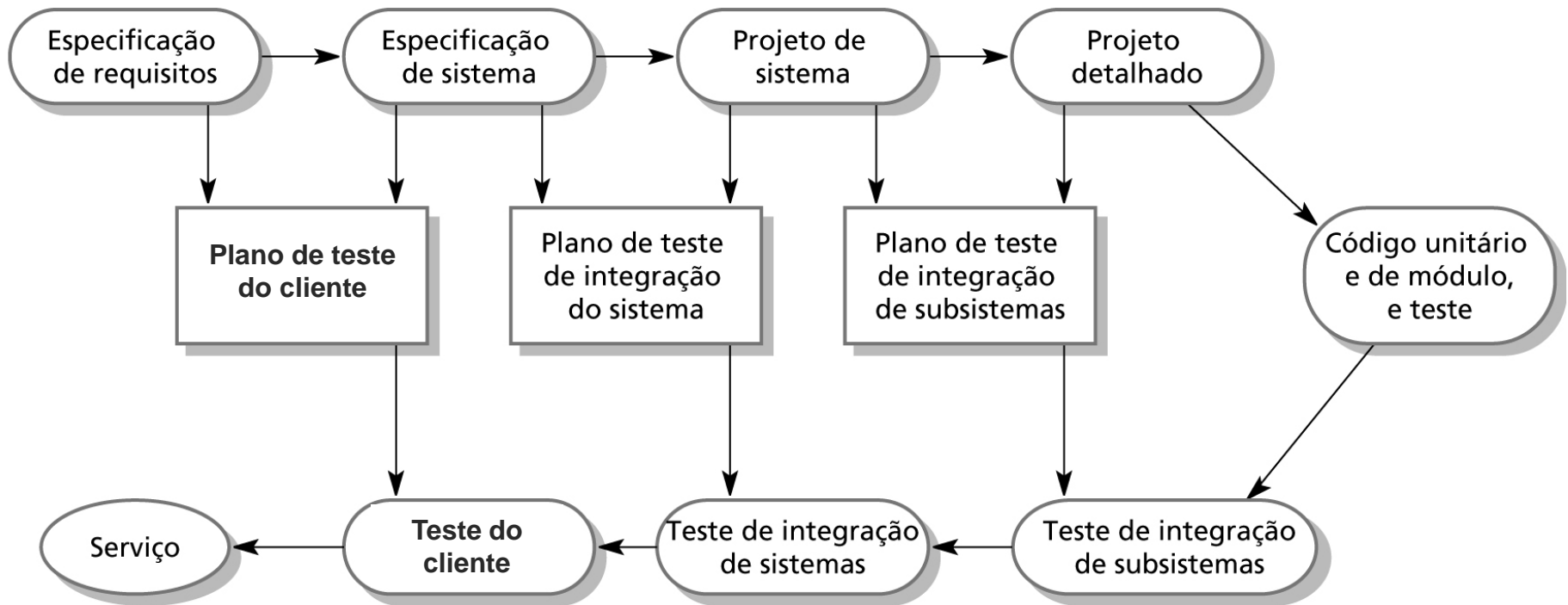
- Cada incremento é testado enquanto é desenvolvido
 - Testes são baseados nos requisitos do incremento
- Em XP, os testes são escritos antes de se iniciar a implementação
 - Desenvolvimento Dirigido por Testes (TDD)



[Desenvolvimento Tradicional]

- 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]



Características do 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

[Teste de Usuário (Cliente)]

■ Teste Alfa

- Grupo de usuários colabora com a equipe de desenvolvimento para testar
- É feito até que o cliente aceite que o sistema seja entregue

A stylized, orange Greek letter alpha symbol (α) is positioned to the right of the text for the Alpha test phase.

■ Teste Beta

- Entrega o sistema a um conjunto maior de usuários (potenciais clientes)
- Os usuários reportam os erros encontrados aos desenvolvedores

A stylized, pink Greek letter beta symbol (β) is positioned to the right of the text for the Beta test phase.

[Bibliografia]

- Ian Sommerville. **Engenharia de Software**, 10^a Edição. Pearson Education, 2019.
 - Seção 2.2.3 Validação do Software