

PÓS-GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

Planejamento e Realização de Teste de Software

Aula 04 Estratégia e Técnicas de Testes

Prof. Rogério Messias

Pauta da aula

- › Correção do exercício
- › IEEE 829 - Padrão para documentação de teste de software e sistema
- › Técnicas de Teste
- › Reflexão de teste ágil de software

Exercício da aula anterior

Elaboração dos casos de testes

		Tipo de CT			Importância		
	Número de CT	FB	FA	FE	Alta	Média	Baixa
UC01	61	1	8	52	18	26	17
UC02	42	1	8	33	2	27	13
UC03	10	1	1	8	8	2	0
UC04	7	1	5	1	6	1	0
UC05	6	1	1	4	0	4	2
Total	126	5	23	98	34	60	32

Estimando o esforço para execução de testes

Estimando que cada caso de testes dure cerca de 15 minutos, podemos chegar à estimativa abaixo

		Tipo de CT			Importância		
	Número de CT	FB	FA	FE	Alta	Média	Baixa
UC01	15,25	0,25	2	13	4,5	6,5	4,25
UC02	10,5	0,25	2	8,25	0,5	6,75	3,25
UC03	2,5	0,25	0,25	2	2	0,5	0
UC04	1,75	0,25	1,25	0,25	1,5	0,25	0
UC05	1,5	0,25	0,25	1	0	1	0,5
Total	31,5	1,25	5,75	24,5	8,5	15	8

Proposta para priorização de escopo de testes

A estimativa pode mesclar as expectativas de risco, prazo e esforço, para determinar quais os casos mais importantes que podem ser realizados no período determinado.

	Prob. Falha	Impacto	Risco
UC01 – Manter Cliente	3	1	3
UC02 – Manter Filme	1	1	1
UC03 – Emprestar Filme	3	3	9
UC04 – Devolver Filme	2	3	6
UC05 – Gerar Relatório	2	1	2

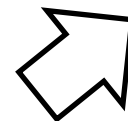
Critérios de aceite para o Plano de Testes

Após identificar os casos de teste, torna-se importante a sua priorização, pois contribuem para a definição de critérios de aceite de forma quantitativa.

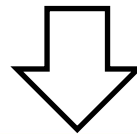
O critério de aceite condiciona um número mínimo de sucesso na execução dos casos de testes.

Importância	Critério de aceite
Alta	100%
Média	70%
Baixa	50%

Recursos necessários



Vídeo Locadora

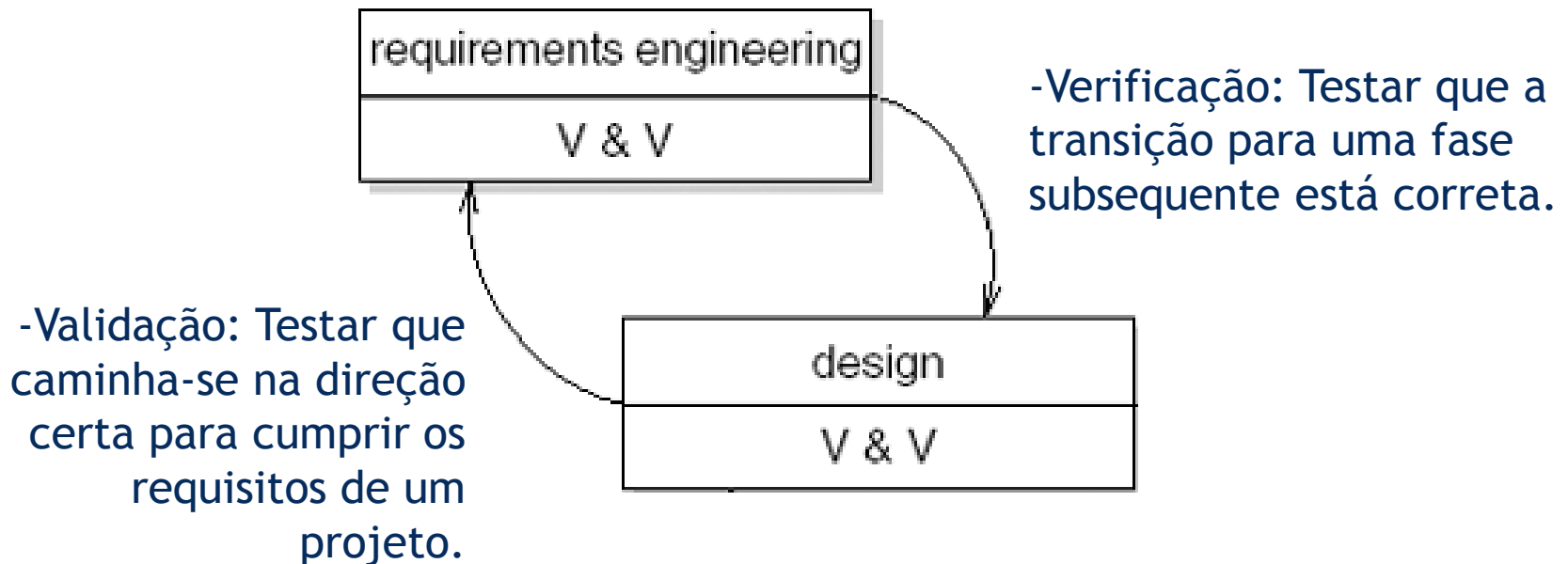


Riscos e contingência

ID	Descrição	Possibilidade	Impacto	Risco	Contingência
1	Indisponibilidade de acesso a internet durante a execução dos testes	1	3	3	Confirmação de disponibilidade de acesso. Modem para acesso à internet 3G
2	Indisponibilidade de pessoas qualificadas para a execução dos testes	2	3	6	Definição de processo de testes. Plano de capacitação pessoal.
3	Indisponibilidade do Google Apps	1	2	2	Uso de stub de testes.
4	Indisponibilidade do Webservice Correios	1	2	2	Uso de stub de testes.
5	Indisponibilidade do servidor de e-mail	1	2	2	Uso de stub de testes.
6	Indisponibilidade de massa de dados	2	2	4	Cadastrar por meio de script SQL filmes e clientes que permitam simular o empréstimo e devolução de filmes.

Quais as críticas da especificação?

- › Estaria a especificação incompleta ou incoerente?



IEEE 829 - Padrão para documentação de testes de Software e Sistemas

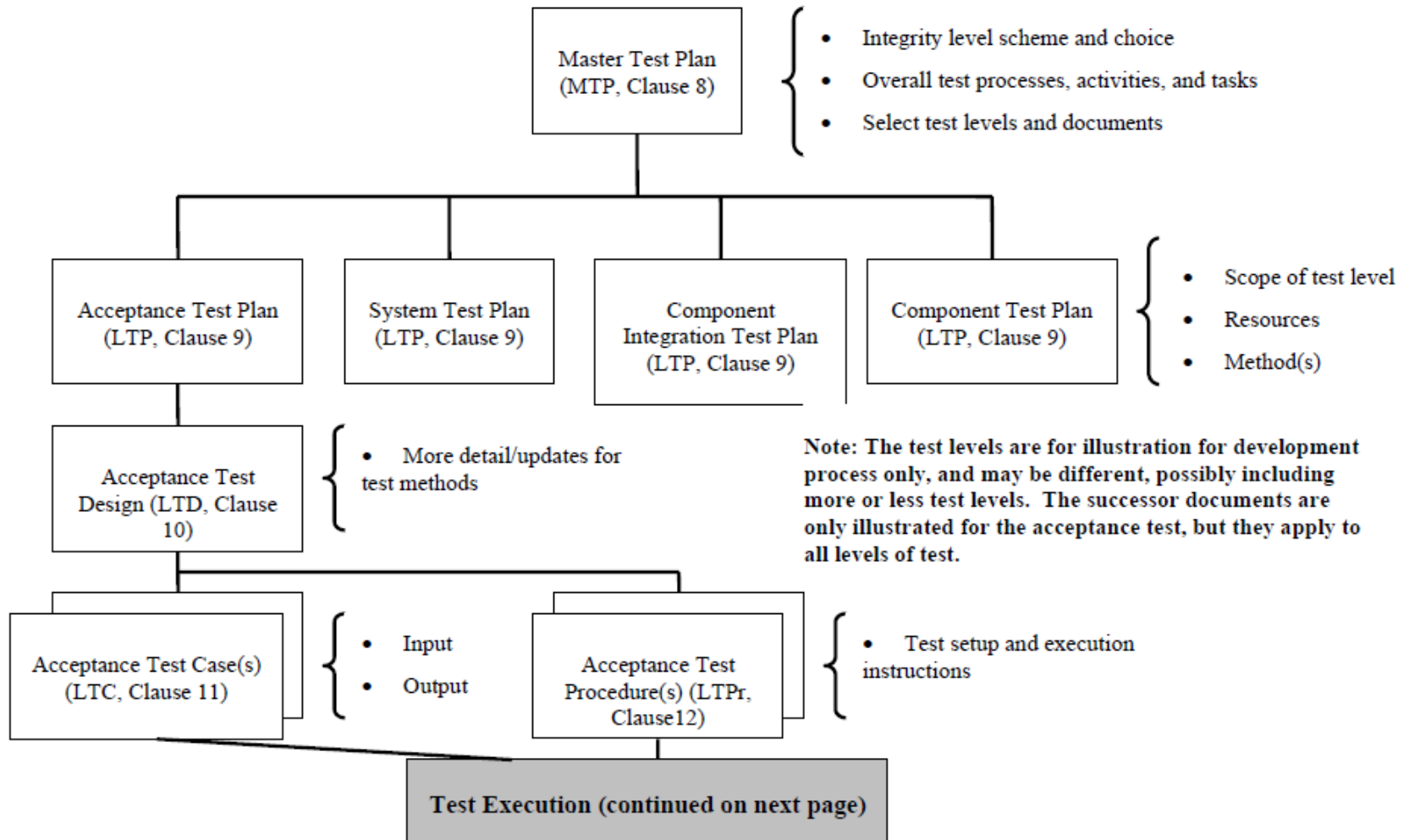
- › Estabelecer um guia para processos, atividades e tarefas de testes que apoiem todos os processos do ciclo de vida de software, incluindo processos de aquisição, fornecimento, desenvolvimento, operação e manutenção.
- › Definir tarefas de testes, entradas e saídas requeridas.
- › Definir o uso e o conteúdo dos documentos de testes:
 - › Projeto de testes
 - › Casos de Testes
 - › Procedimentos de Testes
 - › Relatórios de anomalias
 - › Relatórios de testes
 - › Relatórios de níveis de testes
 - › Relatório de testes mestre

Atividade desenvolvida

- Corresponde a 80% do Plano de Testes Mestre.
- Atividade desenvolvida durante a especificação de requisitos do projeto.
- Determinou as diretrizes do planejamento da execução dos testes do sistema.
- Estabeleceu quais os critérios de aceite do sistema.
- Identificou riscos.
- Determinou um plano de ação para mitigá-los.

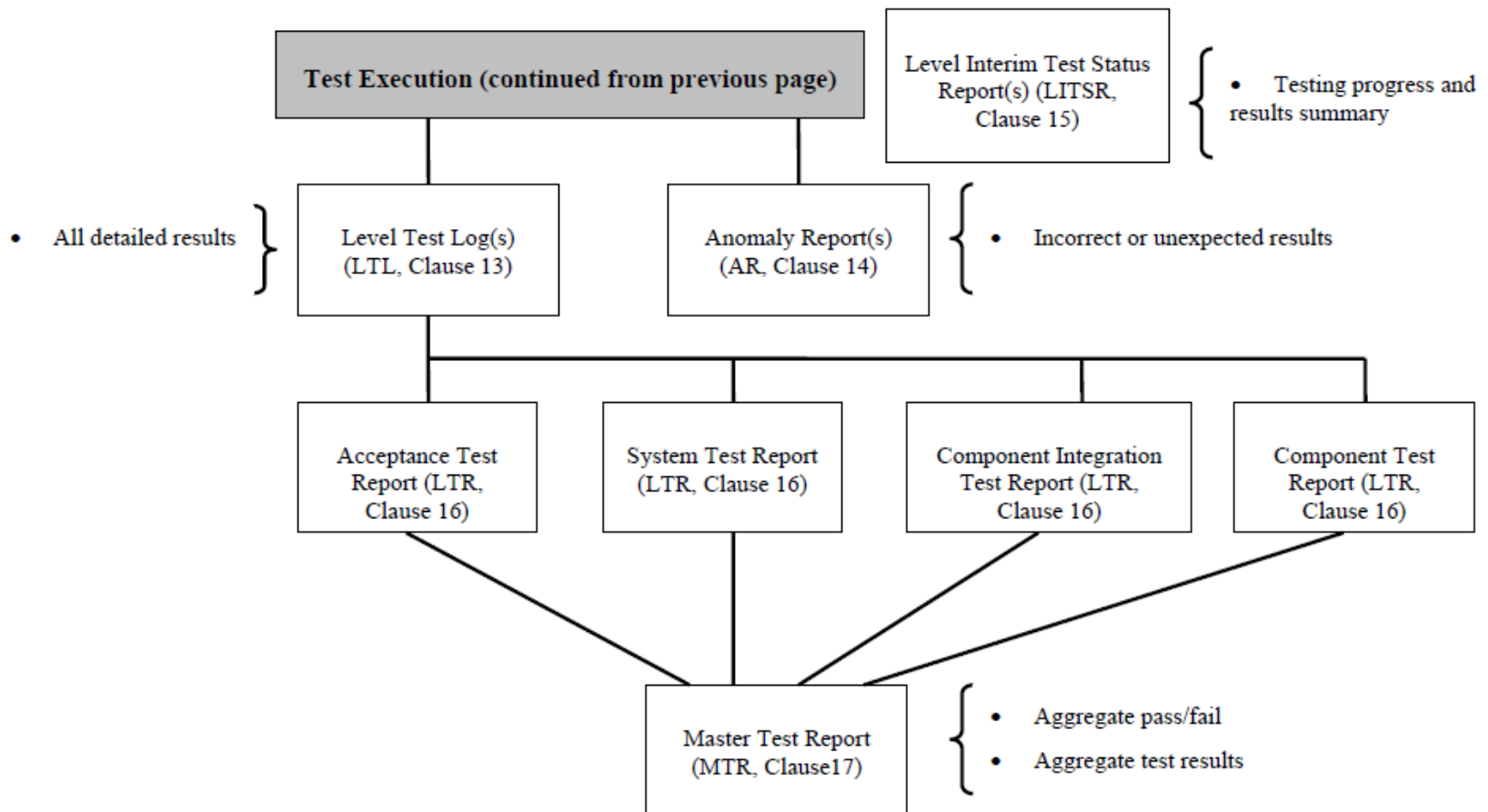
Visão geral dos documentos de planejamento

Figure 7-1 Test Documentation Overview (informative)

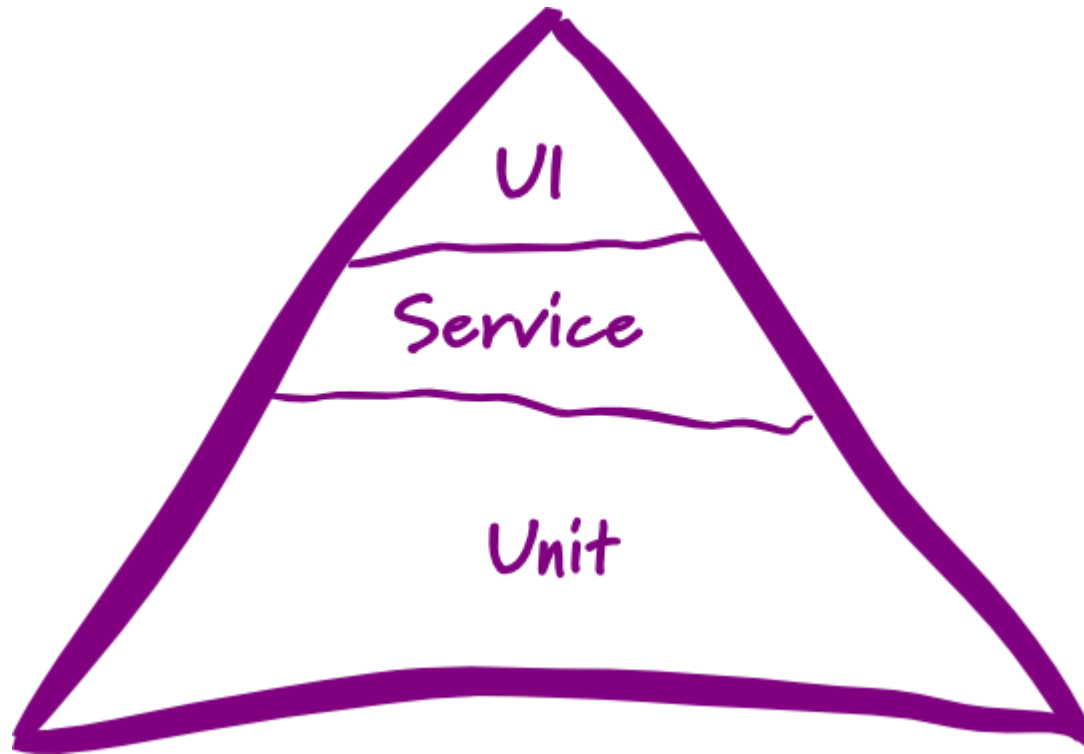


Visão geral dos documentos de resultados

Figure 7-1 Test Documentation Overview (informative)
(continued)




Visão de testes ágeis



Martin Fowler, 2012

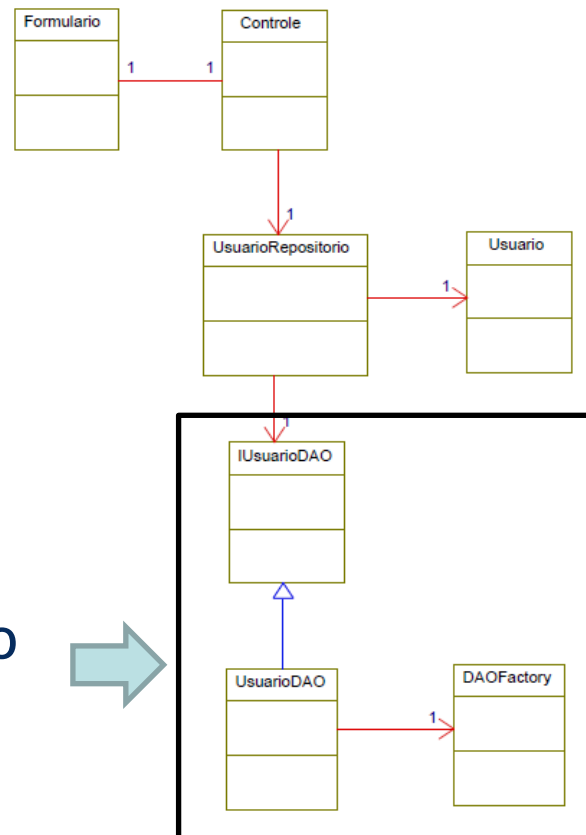
Atividades de testes associadas às fases de desenvolvimento

Fase	Atividade
Engenharia de Requisitos	<ul style="list-style-type: none">• Determinar a estratégias de testes• Especificação de requisitos de testes• Geração de dados de testes funcionais
Projeto	 <ul style="list-style-type: none">• Checar a consistência entre a especificação de requisitos e o projeto (design)• Avaliar a arquitetura de software• Testar o projeto de arquitetura• Geração de dados de teste funcional e estrutural
Implementação	<ul style="list-style-type: none">• Verificar a consistência entre o projeto (design) e a implementação• Testar o programa
Manutenção	<ul style="list-style-type: none">• Repetir os testes acima de acordo com o nível de retrabalho.

Fonte: VLIET, 2008

Teste de Componente

- Visão estrutural das classes que realizam o UC01 Cadastra Cliente



Componente que realiza a abstração da camada de persistência.

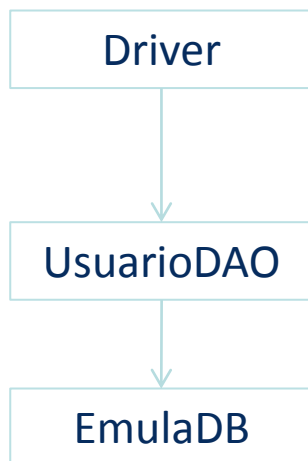
Teste de Componente - Especificação

- Análise da Especificação de componentes (ECO) considerando cada função a ser testada

Identificação	UsuarioDAO
Tipo	Componente de acesso a dados
Propósito	Encapsular os mecanismos de acesso a dados (conexão com o banco de dados e queries SQL)
Função	Manipular as informações de clientes no banco de dados
Subordinados (composição)	A funcionalidade é realizada pela classe UsuarioDAO que implementa a interface IUsuarioDAO e pela classe DAOFactory que estabelece a conexão com o banco de dados.
Dependência	O componente UsuarioDAO depende da classe DAOFactory e do driver de acesso a dados MySQL
Interface	<p>consulta(String id) – recebe como parâmetro o id do usuário e retorna a senha, nome e perfil</p> <p>cadastra (String id, String nome, String senha,String p) – recebe como parâmetros os dados do usuário e retorna 0 erro usuário não cadastrado ou 1 usuário cadastrado com sucesso.</p> <p>exclui (String u) – recebe como parâmetro o id do usuário a ser excluído retorna 0 – usuário não excluído ou 1 usuário excluído com sucesso.</p> <p>altera(String id,String nome, String perfil) – recebe como parâmetro o id do usuário a ser alterado, o nome e perfil a ser alterado, retorna 0 – usuário não alterado ou 1 usuário alterado com sucesso.</p> <p>liberacaoAcesso(String id, String senha) – recebe como parâmetro o id e a senha, retorna as funções que o usuário tem direito de acesso</p> <p>alteraSenha(String id, String senhaAtual, String novaSenha) – recebe como parâmetro o id e senha atual do usuário e a nova senha.</p>
Recursos externos (impressora, banco de memórias, bibliotecas matemáticas)	
Processamento (refina o atributo função)	Para manipular os dados da queries SQL são pré-compiladas para reduzir o tempo de execução.
Dados	<p>Id – código de identificação do usuário</p> <p>Senha – senha de acesso</p> <p>Nome – nome do usuário</p> <p>Perfil – aplicações liberadas para o usuário.</p>

Técnicas de Teste de Unidade

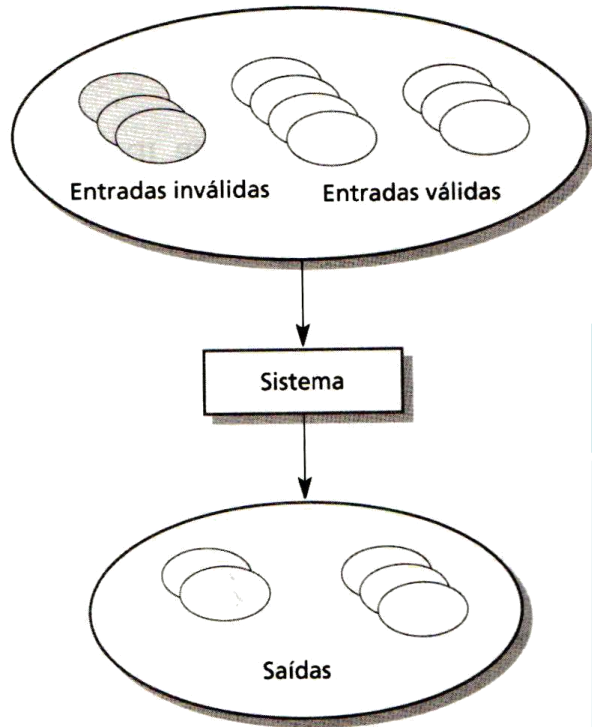
- › Técnica de caixa preta: classes de equivalência e análise do valor limite
- › Técnica de caixa branca: teste do caminho básico (todos os desvios serão testados para condições verdadeiras ou falsas)



Teste de unidade

```
public interface IUsuarioDAO {  
    public ResultSet consulta(String id);  
    public int cadastra (String id, String nome, String senha,String p);  
    public int exclui (String u);  
    public int altera(String id,String nome, String perfil);  
    public ResultSet liberacaoAcesso(String id, String senha);  
    public int alteraSenha (String id, String senha, String novaSenha);  
}
```

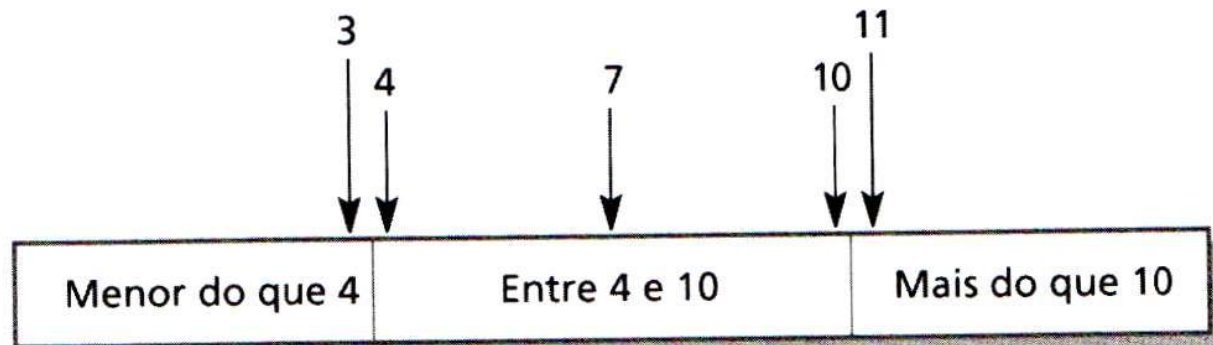
Classes de Equivalência



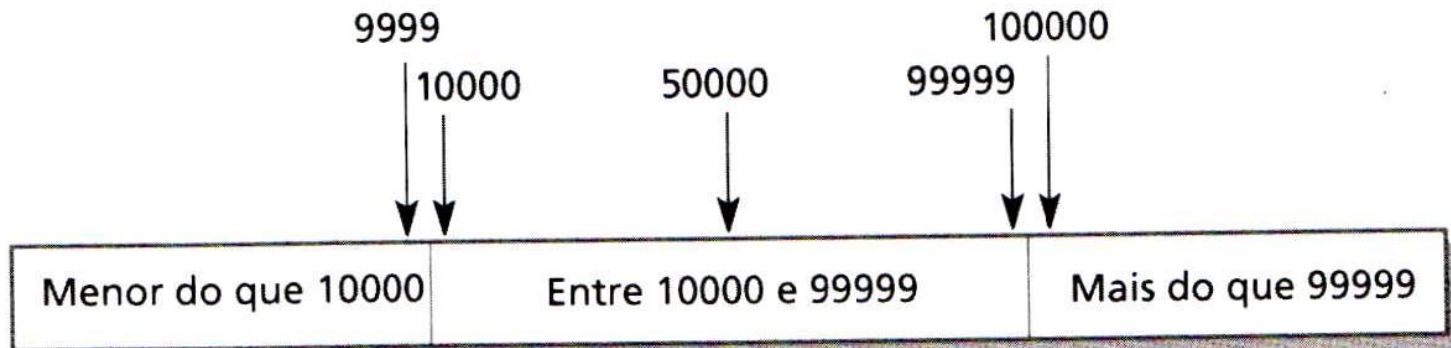
Sommerville, 2008

Entradas	Classes válidas	Classes inválidas
Dados do usuário	Conjunto de dados validos para usuário Conjunto de dados validos de senhas com 1 caractere Conjunto de dados validos de senhas com 7 caracteres	Conjunto de dados id, senha, nome, perfil com informações vazio ou nulo Conjunto de senhas com mais de 8 caracteres Conjunto de perfis não cadastrados Conjunto de usuários já cadastrados

Análise de valor limite



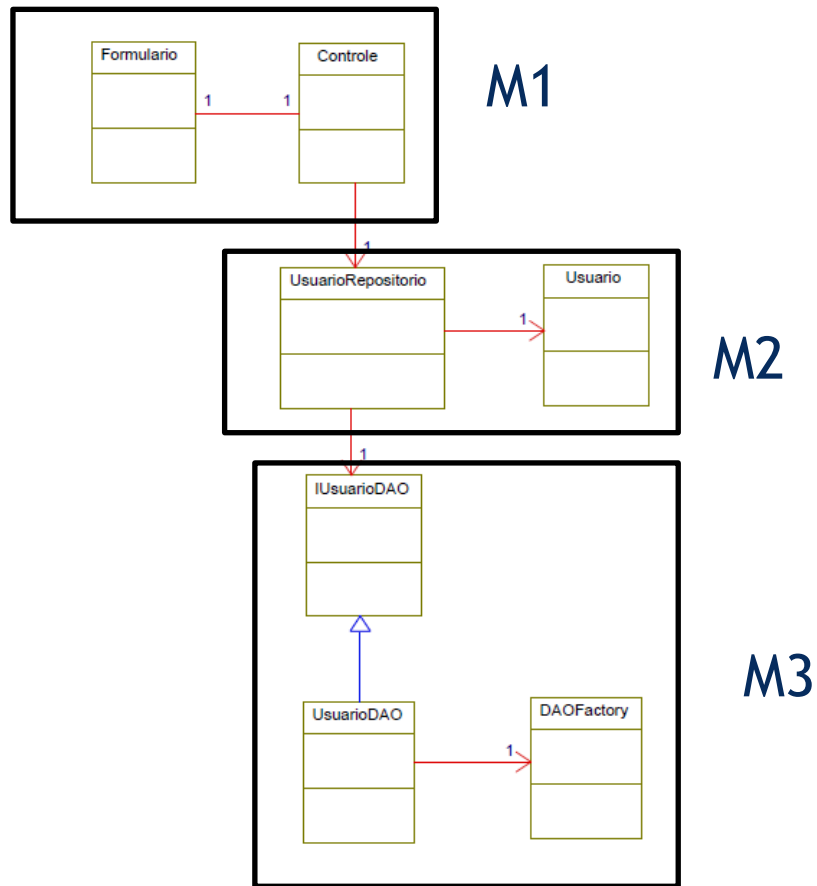
Número de valores de entrada



Sommerville, 2008

Teste de Integração

Visão estrutural da arquitetura



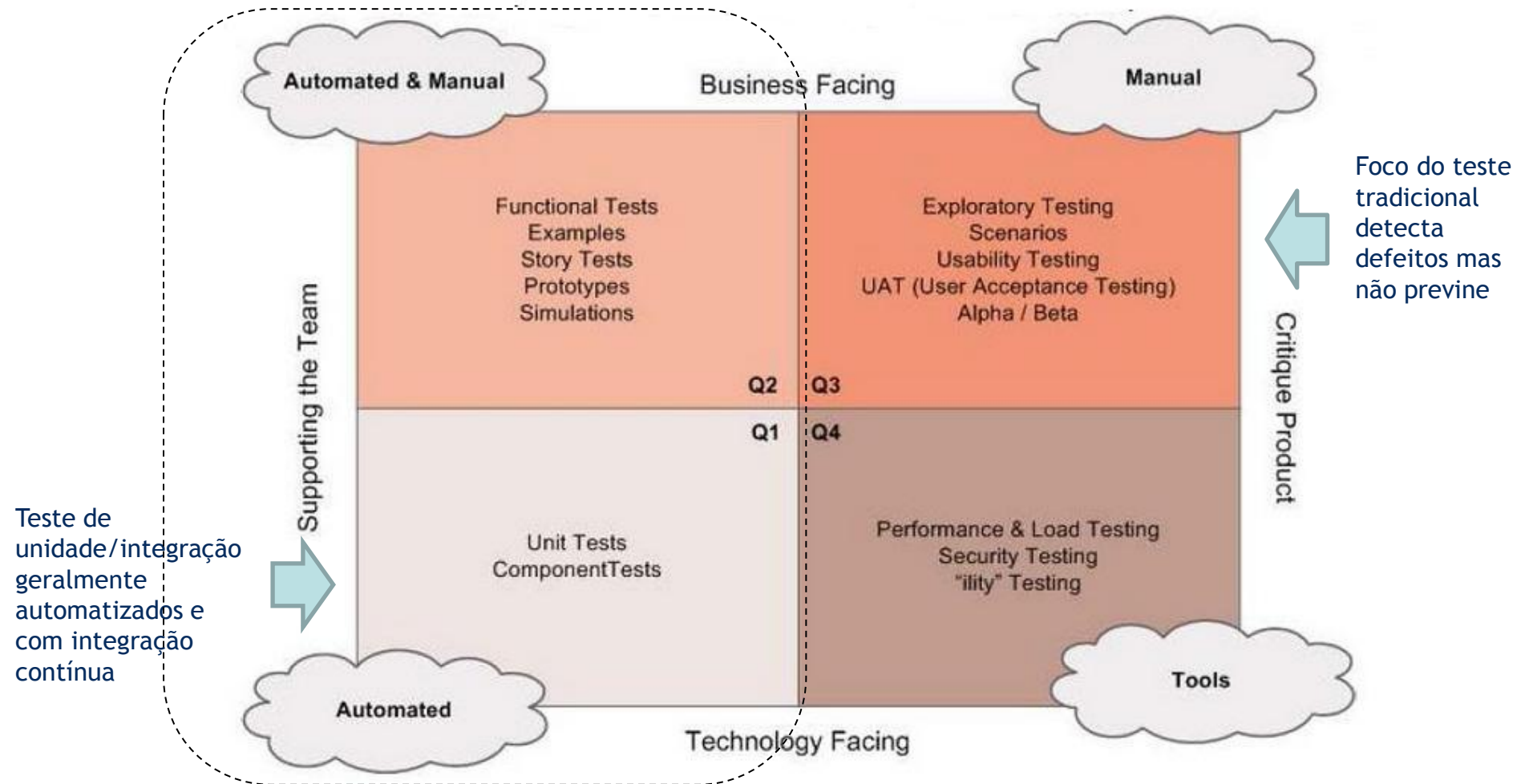
Seqüência de
integração-
M3,M2,M1

Documentação de testes de unidade e integração

Method Summary	
void	CT01CadastraUsuario dados invalidos() id: CT01CadastraUsuario_dados_invalidos Objetivo:verificar o comportamento da funcao cadastra usuario com dados em branco Pré-condições: o banco de dados nao valida campos em branco
void	CT02CadastraUsuario dados invalidos() id: CT02CadastraUsuario_dados_invalidos Objetivo:verificar o comportamento da funcao cadastra usuario com senha maior que 8 caracteres Pré-condições: o banco de dados valida senha com mais de 8 caracteres
void	CT03CadastraUsuario dados invalidos() id: CT02CadastraUsuario_dados_invalidos Objetivo:verificar o comportamento da funcao cadastra usuario com dados nulo Pré-condições: o banco de dados valida campos null
void	setUp()
void	tearDown()

Utilização do próprio código fonte para realizar a documentação, identificando as condições necessárias para a execução do teste

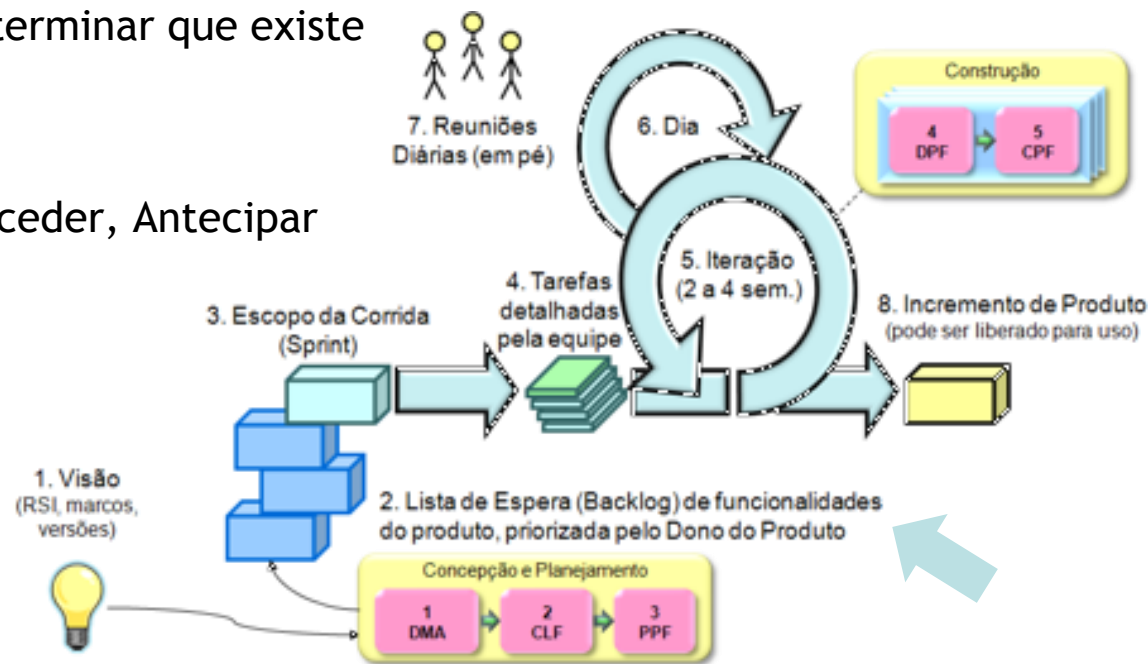
Desenvolvimento Ágil e Testes



Lisa Crispin, Janet Gregory, Agile Testing: a practical guide for testes na agile teams, Addison Wesley, 2009

Desenvolvimento Ágil e Testes: O que fazemos?

- › Detectamos:
 - › Descobrir, determinar que existe
- › Ou Prevenimos:
 - › Impedir, Anteceder, Antecipar



- › Backlog baseia-se em funcionalidades do produto
- › Incrementos baseiam-se em liberações para o uso

Próxima Aula

- › Entrega de atividade individual - Artigo 02 - Estratégia de teste de Software >> Leitura importante para a realização do trabalho em grupo
- › Apresentação do trabalho em grupo

Referências

IEEE 829. Standard for software and System test documentation, 2007

FOWLER, Martin. TestPyramid, disponível em <http://http://martinfowler.com/bliki/TestPyramid.html> acessado em 25/08/2012

SOMMERVILLE, I. Engenharia de Software, 2008