





Verificação e Validação de software

No Contexto de Desenvolvimento Ágil de Software

Prof. Ricardo Ajax
ricardoajax@unb.br

1



Histórico - Movimento Ágil

- O processo de software tradicional não era eficiente (Larman, 2004).
- Alta taxa de falha na indústria → Descrição não cumpre sua promessa
- Muito documento antes do “vamos ver” → Código
- **Reação: 17 pessoas Aliança ágil: <http://www.agilealliance.org>, Evento ocorrido em Fev/2001.**
- Uma primeira concordância entre os participantes foi que um método ágil deve ser apto a aceitar e tratar as mudanças inerentes a um desenvolvimento de software (Kock, 2004).
- A partir de então foram estabelecidos valores e princípios que devem estar presente em processos de desenvolvimento que se propõem a ser ágeis.

• **Os quais são:**

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

2

UnB Gama
O novo endereço da Tecnologia.

Movimento Ágil –Valores

- O movimento ágil começou com a publicação de quatro valores e doze princípios.
- Os valores são:
 - **Indivíduos e iterações** acima de ferramentas e processos;
 - **Código funcionando** acima de documentação;
 - **Colaboração do cliente** acima de negociação contratual;
 - **Responder a mudanças** acima de seguir um plano;

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

3

UnB Gama
O novo endereço da Tecnologia.

Movimento Ágil –Princípios

- Satisfazer o cliente**, por meio de entregas constantes e incrementais;
- Receber os pedidos de **mudanças**;
- Entregar software funcionando** com frequência;
- Estabelecer **software funcionando**, pois **é a principal medida** de progresso;
- Desenvolvedores e responsáveis pelo negócio devem **trabalham perto no dia-a-dia** do desenvolvimento (colaboração entre papeis);
- Construir projetos ao redor de **indivíduos motivados**;
- Estabelecer **rotinas de interação pessoal** entre os responsáveis pelo desenvolvimento.
- Promover o **desenvolvimento sustentável**, desenvolvimento ágil;
- Prestar atenção contínua para **excelência técnica e bom design**, pois otimiza a agilidade;
- Simplicidade;
- Equipes auto-organizadas**; e
- Regularmente a equipe reflete sobre **como se tornar mas efetiva**.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

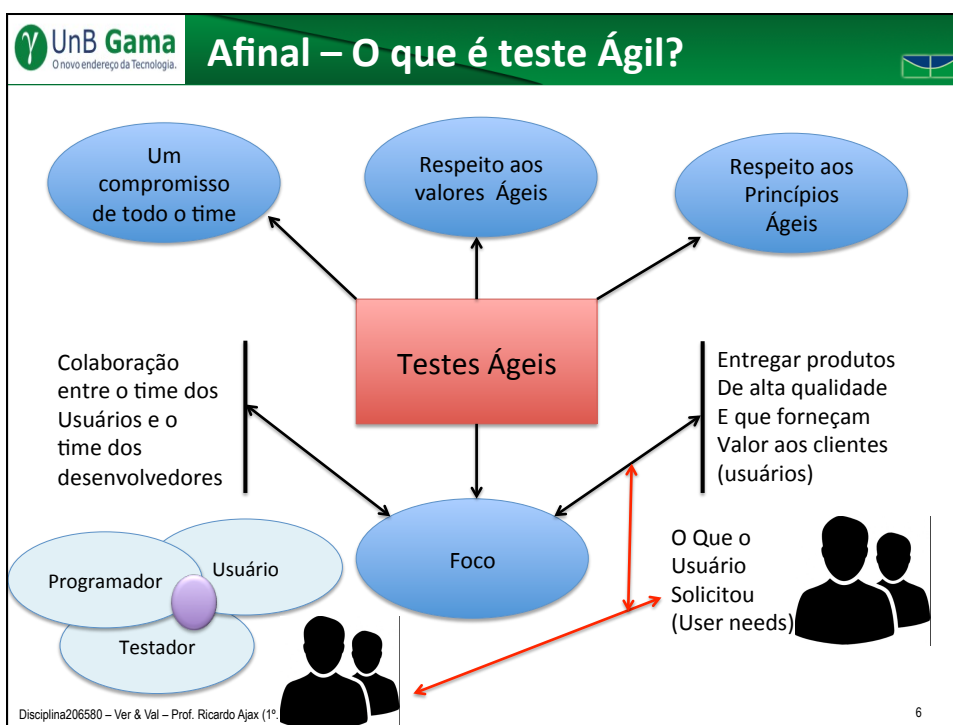
4


UnB Gama
O novo endereço da Tecnologia.

Introdução


- O que significa Testes Ágeis (*Agile Testing*)?
 - TDD, testes de integração de código, testes de unidade de código são práticas ágeis, utilizadas por times ágeis.
 - No entanto, testes ágeis incluem testes que criticam o produto, voltados ao negócio (*business-facing* ou *customer-facing*)
 - Foco em descobrir o que está faltando no produto e como podemos melhorá-lo.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016) 5



**UnB Gama**
O novo endereço da Tecnologia.


Vontando à questão




- O que significa Testes Ágeis (*Agile Testing*)?
 - Inclui testes funcionais, sistema, carga, desempenho, segurança, stress, aceitação....
 - Inclui níveis de teste unitários, integração, sistemas, aceitação
 - Inclui ambientes diversos definidos em comum acordo clientes e desenvolvedores
 - Inclui o uso de técnicas especiais para testar o software ficando pronto passo a passo e rapidamente.
- Todos estes tipos de testes podem ser apropriados para qualquer projeto, sendo um projeto ágil ou usando metodologias tradicionais.
- Testes ágeis não significam apenas testes em projetos ágeis!!

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

7

**UnB Gama**
O novo endereço da Tecnologia.


Testes ágeis x tradicionais




- Tradicionais
 - “Me desculpe, mas os requisitos estão congelados. Espere pela outra *release*”
 - “Porteiros da qualidade”
 - Fases pós-desenvolvimento são esperadas para aumentar a qualidade do código já completado
 - Garantir que todos os requisitos especificados sejam entregues no produto final
 - Testadores estudam os requisitos para escrever os planos de testes e então esperam o produto ser entregue para ser testado

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

8




Testes ágeis x tradicionais




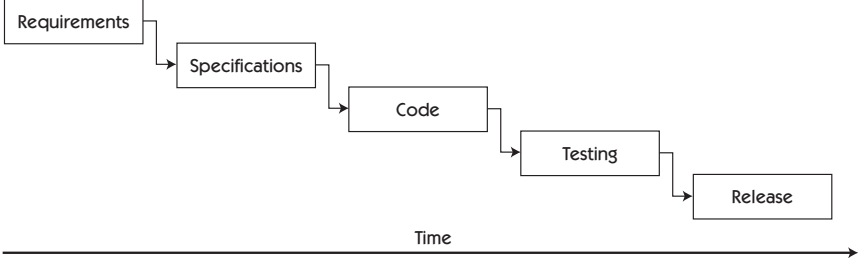
- **Testes Ágeis**
 - Os critérios de qualidade do produto são definidos pelo cliente e não pela equipe de qualidade
 - O papel de um profissional de teste em uma equipe ágil é:
 - auxiliar o cliente a definir qual o valor de cada iteração e escrever testes que garantam que ele obtenha esse valor
 - conduzir testes exploratórios manuais
 - parrear com programadores para melhorar os testes
 - Equipe tem entendimento detalhado dos requisitos
 - Foco em entrega de valor
 - Testadores procuram por maneiras de contribuir ao longo do ciclo de vida
 - *Whole-team approach*

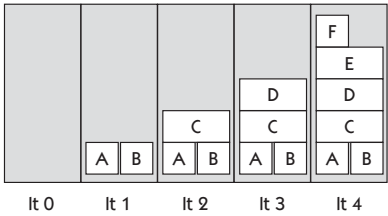
Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)
9



Testes ágeis x tradicionais









Agile:
Iterative & incremental



- Each story is expanded, coded, and tested
- Possible release after each iteration

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)
10


O Que muda? Alguns conceitos que inovaram




- Refatoração (Fowler 1999) – forma disciplinada de fazer pequenas mudanças no código fonte sem alterar o comportamento, enquanto se aperfeiçoa o design;
- Desenvolvimento Dirigido a Testes (TDD – Beck 2003, Astels 2003) – uma abordagem incremental e iterativa da programação, onde se tem em vista primeiro gerar os testes e por conseguinte, criar a funcionalidade;
 - Antes: os testes eram consequência
 - Em TDD, a funcionalidade é a consequência
- Testes substituem artefatos tradicionais (Ambler, Scott) – testes substituem os artefatos, pois testes de aceitação comprovam se a funcionalidade foi aceita e está de acordo com os requisitos do projeto. Com a perspectiva do TDD os desenvolvedores criam os testes antes das funcionalidades, assim eles pensam no que o código fonte deve fazer antes de ser escrito;

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)
11


O Que muda?


- Modelo ágil de desenvolvimento – Devido à interação próxima ao cliente, torna-se reduzida ou **desnecessária a necessidade de criação de extensa documentação**, assumindo-se que o cliente estará indicando o caminho que deve ser seguido pelos desenvolvedores. Assim, criam-se novos modelos que auxiliam no desenvolvimento.
- **Qualidade** focada no **produto** gerado \leftrightarrow Necessidades de novas técnicas, medições e ferramentas (aprofundamento das características de qualidade do produto);
- Se tudo deve ser rápido e correto, **automação** das atividades de testes é um bom tema de pesquisa no contexto de desenvolvimento ágil, assim como são requisitos não funcionais relacionados à qualidade esperada pelo cliente para o produto gerado.



Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)
12

**Então...Novas abordagens - SCRUM**

- A metodologia de gerenciamento SCRUM é framework leve com ampla aplicabilidade para gerência e controle iterativo e incremental dos projetos.
- Papel crítico: No SCRUM “*Product Owner*” trabalha muito perto da equipe de desenvolvedores para auxiliar na identificação e priorização das funcionalidades do sistema na forma de “*Product Backlog*”.
- O “*Product Backlog*” consiste de *features*, correções de defeitos, requisitos não funcionais, etc.
 - O que for preciso para alcançar sistema de software funcionando.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)



13

**Então...Novas abordagens - SCRUM**

- A metodologia de desenvolvimento SCRUM tem como principais vantagens:
 - a rapidez no desenvolvimento e
 - a satisfação do cliente - devido a constante entrega incremental do software.
- Equipes pequenas, auto organizadas, altamente motivadas e comunicação contínua entre cliente e equipe de desenvolvimento são aspectos fundamentais a serem considerados.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)



14

**Então...Novas abordagens - Lean + Kanban**

- A metodologia Lean de desenvolvimento tem seus princípios e práticas no movimento Lean Enterprise, e as práticas de companhias como a Toyota.
- Essa metodologia é focada na equipe, agregando valor ao produto ou serviço a ser entregue ao consumidor.
- Lean elimina o supérfluo por meio de práticas que selecionam somente as *features* que são realmente importantes para o projeto.
 - Foca em velocidade e eficiência do desenvolvimento do fluxo de trabalho, e depende do rápido e confiável *feedback* entre programadores e consumidores.
- A Metodologia *Kanban* de gerenciamento, baseada na entrega contínua, enquanto não sobrecarrega o time de desenvolvimento. Como o SCRUM, o Kanban foi projetado para auxiliar o time de desenvolvimento a trabalhar unido com maior desempenho.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)



15

**Então...Lean + Kanban – Principais Vantagens**

- Prevenção de grande carga de trabalho,
- desenvolvimento flexível,
- redução dos custos,
- minimizar o tempo de espera e os custos de logística,
- redução dos níveis de estoque e custos de overhead.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)



16

**Então...Lean + Kanban - Desvantagens**

- No método *Lean* dependendo da velocidade do *feedback*, a qualidade esperada pode não ser alcançada.
- As desvantagens de se utilizar o *Kanban* são a constante manutenção do tamanho conforme as necessidades de mudança, pode ser um processo lento e difícil de gerenciar.
- Perda de cópias tags/formulários e cartões do Kanban. Utilização desse método reduz a oportunidade de trabalhar muito a frente do planejado, colocando com isso pressão na equipe.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)


17

**Então...Novas abordagens – XP Extreme Programming**


- A metodologia Extreme Programming (XP) é uma aproximação disciplinada para entrega rápida e contínua de software de alta qualidade.
- Promove um grande envolvimento do consumidor, *loops* de *feedbacks* rápidos, teste e planejamento contínuos, assim como trabalho em equipe para entrega de um software funcionando em intervalos frequentes (e os menores possíveis).
- A receita original da metodologia XP baseia-se em quatro princípios básicos: simplicidade, comunicação, feedback e coragem.
- Juntamente com 12 práticas de suporte: planejamento, pequenas releases, testes de aceitação, design simples, programação em pares, desenvolvimento baseado em testes, refatoração, integração contínua, autoria de código coletiva, passo sustentável e padrões de código.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

18




Então...XP - Principais Vantagens




- Entrega contínua de código para o cliente,
- grande quantidade de código,
- código de alta qualidade e
- acompanhamento de perto pelo cliente.
- Mas, se o cliente acompanha de perto, é interessante que ele também:
 - Tenha um certo conhecimento técnico que lhe favoreça acompanhar a equipe
 - Seja capaz de avaliar os resultados dos testes na medida em que eles vão sendo feitos e apresentados
 - Seja capaz de raciocinar junto com a equipe quanto a decisões técnicas (pois as negociais ele deve conhecer e ter capacidade de decidir após rápidas consultas aos usuários negociais) → Um super cliente?

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

19





Então...XP – Principais Desvantagens



- Os principais problemas ocorrem quando os conceitos não são aplicados corretamente no processo de desenvolvimento, por exemplo, quando os clientes não se comprometem com a entrega rápida de *feedbacks*.
- Outro problema que pode ocorrer é causado por grande envolvimento e poder do cliente, o que pode acarretar necessidades durante o processo de produção.
- Sob a ótica de uma empresa, o ideal de programação em par diminui a produtividade, pois um computador está sendo utilizado por dois programadores.
 - Mas TQM avisa: Fazer correto desde o início economiza tempo (e dinheiro)

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)



20

**Então...Novas abordagens – Família Crystal**

- Crystal é uma família de metodologias ágeis como Crystal Clear, Crystal Yellow, Crystal Orange e outras.
- Essa é uma das aproximações mais adaptáveis. Seus fatores dependem do tamanho das equipes, quão crítico é o sistema e as prioridades do sistema.
- A família Crystal trata da realização de cada projeto de uma forma diferente, pois segundo a metodologia cada um desses necessita de uma composição diferente de políticas, práticas e processos para se adequar às características únicas do projeto.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)


21

**Então...Novas abordagens - DSDM**


- DSDM (Dynamic Systems Development Method) fundamenta-se nos seguintes princípios:
 - negócio, necessidade, valor, envolvimento do usuário ativo, equipes providas com poder, entregas frequentes, teste integrado e colaboração dos *stakeholders*.
- DSDM especificamente sustenta que o propósito de negócio é o principal critério para entrega e aceitação do sistema.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

22




Então...Novas abordagens - DSDM




- Os requisitos são primeiramente tratados em alto nível no início do projeto.
- O retrabalho é embutido no processo, e todas as medidas no projeto devem ser reversíveis.
- Os requisitos são planejados e entregues rapidamente e priorizados segundo as regras *MoSCoW*:
 - *M – Must have requirements*
 - *S – Should have if at all possible*
 - *C – Could have but not critical*
 - *W – Won't have this time, but potentially later.*
- O DSDM é independente e pode ser implantado em conjunto com qualquer outra prática iterativa como XP

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016) 23




Então...Novas abordagens – FDD




- FDD (feature Driven Development) é um processo de pequenas iterações dirigido por modelo. As *features* são pequenos resultados (úteis aos olhos dos clientes).
- A FDD tem como fundamento para as entregas das *features* as oito práticas a seguir:
 - Domínio do Modelo de Objeto;
 - Desenvolvimento pela *features*;
 - Autoria da(o) classe/componente;
 - Inspeções;
 - Gerência de configuração;
 - *Builds* regulares; e
 - Progresso e resultados visíveis.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016) 24




UnB Gama
O novo endereço da Tecnologia.

Então...FDD – Principais Vantagens




- Entregas incrementais,
- qualidade e
- resultados visíveis para o cliente.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)
25





UnB Gama
O novo endereço da Tecnologia.

Comparando





Método Ágil	Características	Pontos Importantes
SCRUM	<ul style="list-style-type: none"> - Framework para gerência e controle de todos os tipos de projetos; - Product Owner trabalha muito perto da equipe de desenvolvedores; 	<ul style="list-style-type: none"> - Rapidez no desenvolvimento e a satisfação do cliente; - Equipes pequenas, auto organizadas, altamente motivadas e comunicação contínua entre cliente;
XP	<ul style="list-style-type: none"> - Entrega rápida e contínua de software de alta qualidade; - Loops de feedbacks rápidos, teste e planejamento contínuos; 	<ul style="list-style-type: none"> - 4 princípios básicos: simplicidade, comunicação, feedback e coragem. - Código de alta qualidade e acompanhamento de perto pelo cliente; - Muito poder ao cliente pode ocasionar muita mudanças;

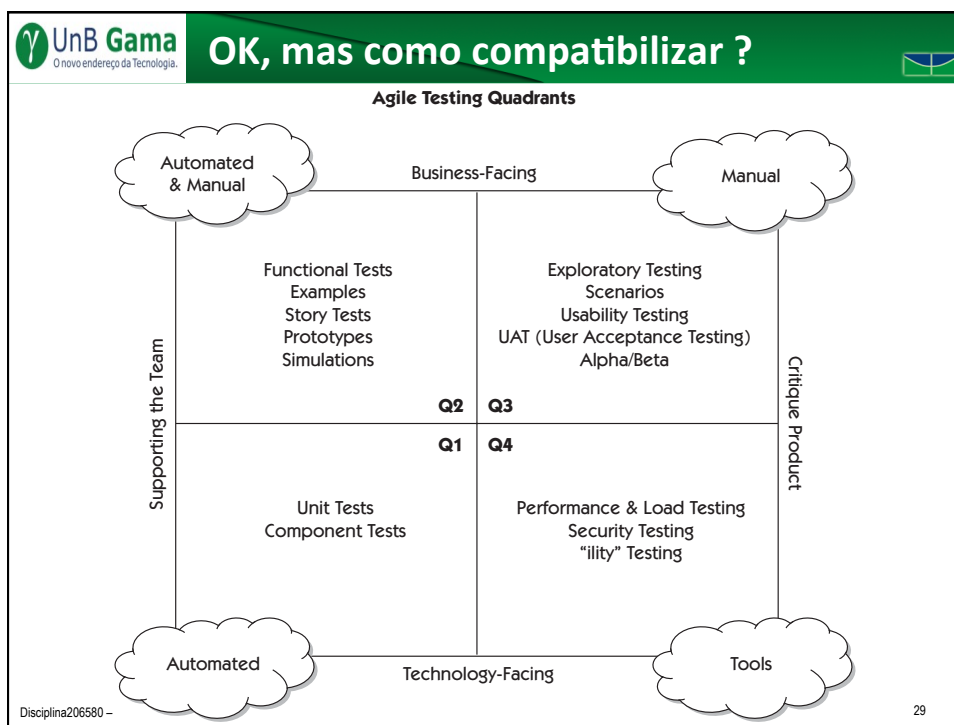
Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)
26

 Comparando 		
Nome	Características	Pontos Importantes
Família Crystal	- Uma família de metodologias ágeis: Crystal Clear, Crystal Yellow, Crystal Orange e outras;	- Trata da realização de cada projeto de uma forma diferente;
LEAN + KANBAN	- Lean segue os conceitos da qualidade total; - Kanban entrega contínua, enquanto sem sobre-carregar o time de desenvolvimento	- No método Lean dependendo da velocidade do feedback, a qualidade esperada pode não ser alcançada - Perda de cópias tags/ formulários e cartões do Kanban. Utilização desse método reduz a oportunidade de trabalhar a frente do planejado
<small>Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)</small>		

27

 Comparando 		
DSDM	- Os requisitos são tratados em alto nível no início do projeto. - Retrabalho é embutido no processo, e todas as medidas no projeto devem ser reversíveis.	- O DSDM pode ser implantado em conjunto com qualquer outra prática iterativa como XP ou RUP.
FDD	FDD é um processo de pequenas iterações dirigido por modelo; Oito práticas: Domínio do Modelo de Objeto; Desenvolvimento pela features; Autoria da(o) classe/ componente; Inspeções; Gerência de configuração; Builds regulares; e Progresso e resultados visíveis.	- Entregas incrementais, qualidade e resultados visíveis para o cliente. - Requisitos estáticos e obsoletos. - Atualmente é necessária uma aproximação mais flexível para se manter competitivo no mercado.
<small>Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)</small>		

28




UnB Gama
O novo endereço da Tecnologia.


Quadrantes do teste

- ☉ Qualidade de software possui várias dimensões
 - Como saber quais tipos de testes devemos aplicar?
 - Como saber que os testes estão feitos?
 - Quem faz quais testes e como?

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016) 30

**UnB Gama**
O novo endereço da Tecnologia.


Quadrantes do teste




- ⦿ Testes que suportam o desenvolvimento
 - ⦿ Suportam o time enquanto desenvolvem o produto
 - ⦿ Principal diferença para a abordagem tradicional de testes
- ⦿ Quadrante 1
 - ⦿ Desenvolvimento dirigido a testes (núcleo das práticas de desenvolvimento ágil)
 - ⦿ Testes unitários: funcionalidade de pequenos pedaços do software.
 - ⦿ TDD é um bom propósito do Q1

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

31

**UnB Gama**
O novo endereço da Tecnologia.

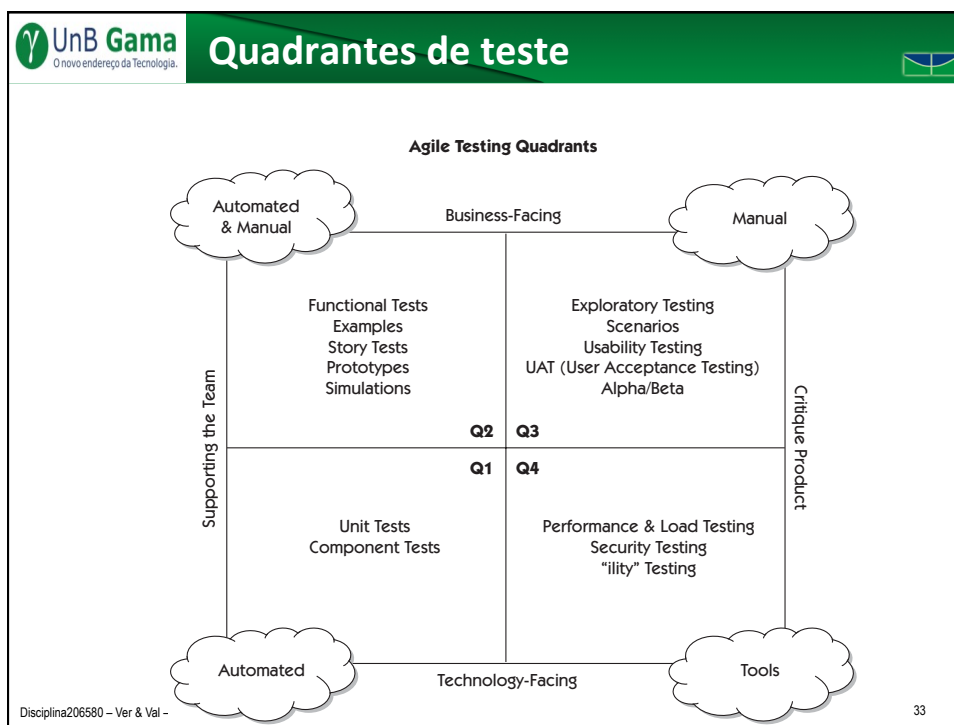
Quadrantes do teste



- ⦿ Quadrante 1
 - ⦿ Qualidade interna: não negociável com cliente, responsabilidade dos desenvolvedores.
 - ⦿ São executados a cada check-in e fornecem feedback contínuo sobre a qualidade interna do código.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

32

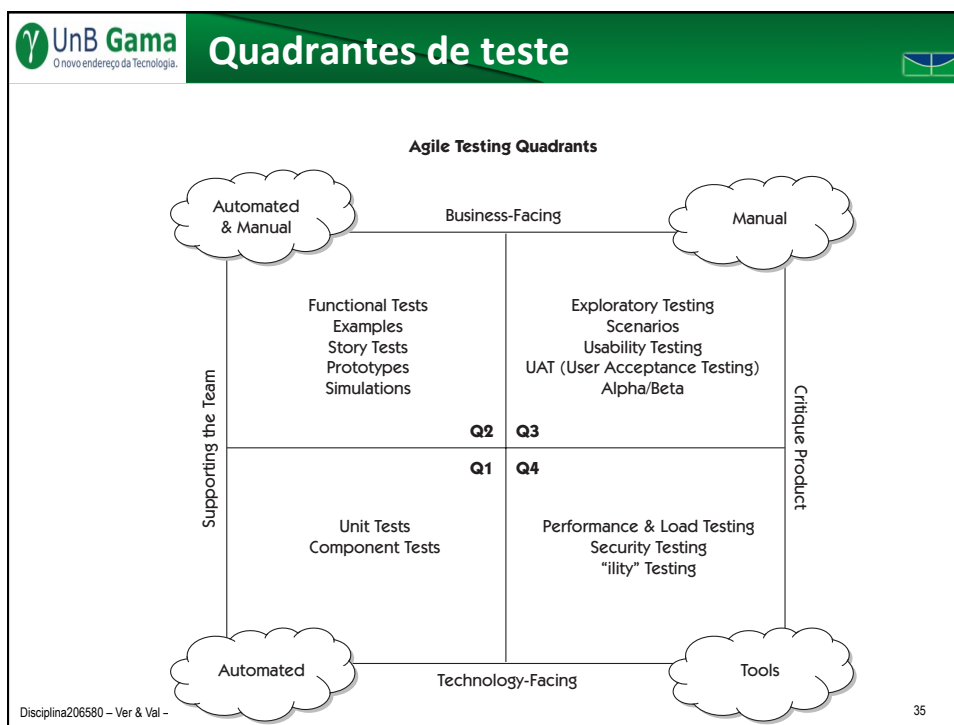


UnB Gama
O novo endereço da Tecnologia.

Quadrantes do teste

- ☉ **Quadrante 2**
 - Suportam o trabalho do time de desenvolvimento, porém em um nível mais alto.
 - Porém, o foco é no que o cliente deseja (business-facing): qualidade externa.
 - Testes derivados do time de clientes: detalhes de cada história, funcional.
 - Quando possível, rodar os testes na camada lógica, sem ter ainda a camada de apresentação.
 - Verifica as interfaces com usuário também
 - Uso de mockups e wireframes para validar a GUI. Não automatizado.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016) 34




UnB Gama
O novo endereço da Tecnologia.

Quadrantes do teste

- ⊙ Testes que criticam o produto
 - ⊙ Dificil expressar os requisitos, até você ver o resultado final.
 - ⊙ É necessária uma revisão do software, com objetivo de apontar melhorias.
- ⊙ Quadrante 3
 - ⊙ Mesmo quando os programadores escrevem um código que passem nos testes voltados ao negócio (business-facing), podem não estar entregando o que o cliente realmente deseja.
 - ⊙ Simular o uso real da aplicação
 - ⊙ Testes manuais, devem ser realizados por pessoas.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016) 36




Quadrantes do teste

- ⦿ Quadrante 3
 - ⦿ Automatizar alguns scripts de dados, por exemplo
 - ⦿ Uso dos sentidos e intuição para checar se o time entregou um valor ao negócio requisitado pelo cliente.
 - ⦿ Testes de aceitação: executado por usuários e clientes. Novas features, mudanças, novas historias. Aprovação das estórias finalizadas.
 - ⦿ Usabilidade: reações do usuário, navegação, itens simples como ordem de tabulação.
 - ⦿ Testes exploratórios: o testador projeta e executa os testes, simultaneamente, usando pensamento crítico para analisar os resultados

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

37

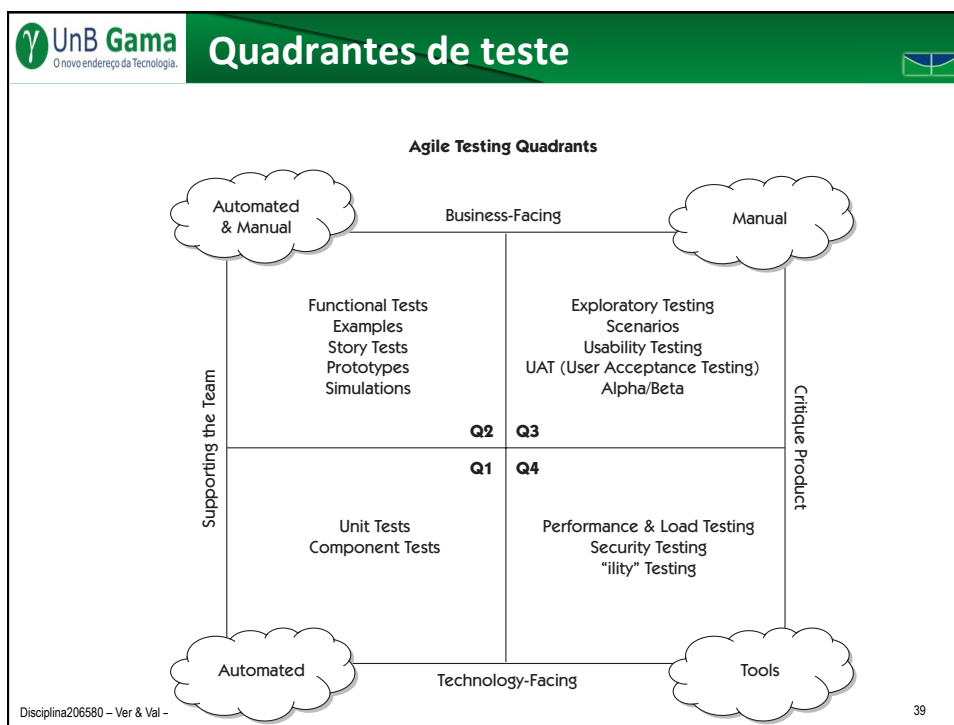


Testes ágeis → Paradigmas

- ⦿ Entender o papel dos testes no contexto de agilidade
 - ⦿ Mais do que descobrir falhas, os testes devem “melhorar o software”
- ⦿ Definir uma matriz de testes com a máxima cobertura possível do software “a ser melhorado”.
- ⦿ Preocupações importantes:
 - ⦿ Integrar os planejamentos dos testes pelas várias sprints
 - ⦿ Qual a documentação mínima para realizar os testes e para registrar seus resultados? Quais os dados relevantes? Como eles devem ser agrupados (Consolidados)? Quais as medições?
 - ⦿ Acompanhar a execução dos testes – Gestão dos testes

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

38




UnB Gama
O novo endereço da Tecnologia.

Quadrantes do teste


⦿ **Quadrante 4**

- ⦿ Testes voltados a tecnologia e que criticam o produto
- ⦿ Necessidade de ferramentas e habilidades específicas
- ⦿ Programadores devem ser capazes, inclusive, de verificar desempenho no nível unitário.
- ⦿ Sempre que possível, devem ser automatizados.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016) 40

**UnB Gama**
O novo endereço da Tecnologia.


Itens importantes




- ⦿ **Quadrante 1 – Unitários e componentes**
 - ⦿ Controle de código fonte e gerenciamento de configuração
 - ⦿ Integração contínua: permite rodar os testes a cada check-in de código.
 - ⦿ Quando o teste falhar, é possível saber QUEM fez a mudança que causou a falha.
 - ⦿ Motiva o programador a rodar os testes antes de “subir” o código.
 - ⦿ A falta desse quadrante resultará em mini cascatas. O testador não terá tempo para testes por causa da baixa qualidade do código.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

41

**UnB Gama**
O novo endereço da Tecnologia.



Itens importantes



- ⦿ **Quadrante 2 – Funcionais, exemplos e protótipos**
 - ⦿ Tais testes se tornarão parte dos testes de regressão
 - ⦿ Nos métodos tradicionais, são gastos alguns meses para levantar os requisitos do software.
 - ⦿ Em ágeis, as estória são escritas pelo time de clientes. Os desenvolvedores podem ajudar a quebrar as estórias em tamanhos mais apropriados.
 - ⦿ Portanto, nem todos os requisitos estão completamente entendidos.
 - ⦿ Os exemplos podem formar a base para os testes e ajudam a entender os requisitos

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

42




Itens importantes


- ⦿ **Quadrante 2 – Funcionais, exemplos e protótipos**
 - ⦿ historia1

Como um comprador, eu quero adicionar itens ao carrinho e então pagar por todos de uma só vez.
 - ⦿ Exemplo 1 para a historia 1


Há cinco itens em uma página. Eu quero selecionar o item 1 de 20,25 reais e colocá-lo no carrinho de compras. Clico para a próxima página, que tem mais cinco itens. Eu selecionei um segundo item na página por 5,38 reais e coloquei no meu carrinho de compras. Quando termino as compras, será apresentado tanto o item da primeira página quanto o item da segunda página no meu carrinho de compras, com o total de 25,63 reais.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)
 43



Itens importantes


- ⦿ **Quadrante 2 – Funcionais, exemplos e protótipos**
 - ⦿ Mostrar qual o código deve ser escrito!
 - ⦿ Pequenas partes do software dão aos clientes a oportunidade de ver a usar a aplicação. Ajustes nos requisitos são esperados.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)
 44

**UnB Gama**
O novo endereço da Tecnologia.


Itens importantes




- ⦿ **Quadrante 3 – Exploratórios, usabilidade e aceitação**
 - ⦿ Difícil de automatizar, pois necessitam de experiência, intelecto humano, instinto.
 - ⦿ Avaliar e criticar o produto significa manipular o sistema em teste e tentar recriar experiências reais dos usuários finais.
 - ⦿ Demonstrar o software ao usuário final para coletar feedback.
 - ⦿ Testar a GUI da aplicação pode ser a melhor maneira de entender a funcionalidade.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

45

**UnB Gama**
O novo endereço da Tecnologia.

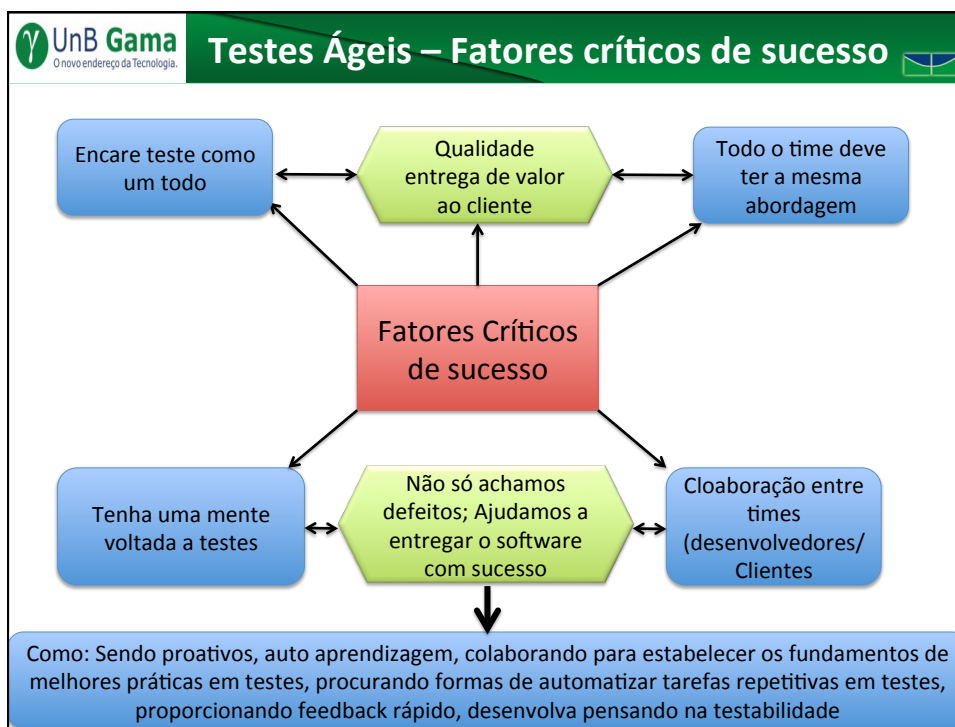
Itens importantes



- ⦿ **Quadrante 4 – Desempenho**
 - ⦿ Requisitos não funcionais.
 - ⦿ O time deve avaliar se possui ou deve adquirir as habilidades necessárias para este tipo de teste
 - ⦿ Vários tipos, tais como manutenibilidade, confiabilidade, compatibilidade, desempenho.
 - ⦿ Verificar se aspectos tecnológicos causam impacto no produto.

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

46



UnB Gama O novo endereço da Tecnologia.

Aliás.... Por que???


Segundo Cock,(2004), automatizar tem alguns benefícios como:

- ◉ Melhorar a produtividade dos testadores
- ◉ Obter uma maior cobertura dos testes (especialmente nos de regressão)
- ◉ Reduzir a duração da fase de execução dos testes
- ◉ Reduzir os custos de manutenção
- ◉ Aumentar a efetividade dos casos de teste

Mas não é tão fácil assim. Algumas perguntas antes de começar.

- ◉ Os casos de testes a serem automatizados estão bem definidos?
- ◉ A infraestrutura e as ferramentas estão prontas?
- ◉ Os profissionais que farão as automatizações têm experiência?
- ◉ Eles conhecem o sistema?
- ◉ A organização está disposta a investir na automação dos testes? Existem benefícios palpáveis?

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)



As top 10 máximas sobre testes ágeis

- Os trabalhos de teste iniciam logo após a obtenção dos requisitos (funcionais ou não funcionais).
- Se o testador não entendeu o requisito → Preocupe-se, pois ele pode não ser testável e é implementável se não é testável?
- Agilidade nos testes não é uma panaceia para resolver todos os problemas. Os problemas existem, os testes apenas os apontam. Mas quem tem de resolver são os integrantes da equipe
- Os responsáveis pelos testes (analistas ou testadores) também fazem parte da equipe
- Início da Sprint ↔ início das atividades de teste.
- Todo o time é responsável pelo resultado do desenvolvimento (inclusive o pessoal de testes).
- As atividades de teste são atividades comuns e corriqueiras da equipe de desenvolvimento
- Nada pessoal, mas o feedback é fundamental para o sucesso do resultado do projeto de software. Forneça feedbacks constantes (para os clientes, para os desenvolvedores).
- O teste em ambiente ágil não é tão diferente assim dos outros ambientes pois, fora seus paradigmas, existem muitas atividades comuns a outros ciclos de vida (teste é um processo).
- Pesquise, pesquise pesquise (ex: ATA – Agile Testing Alliance)

Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

49



Dúvidas?

- Ricardo Ajax
 - Ricardoajax@unb.br



Disciplina206580 – Ver & Val – Prof. Ricardo Ajax (1º. Semestre de 2016)

50