



Promovendo a Colaboração entre Desenvolvedores e Testers

Felipe Albertão, Software Engineer, PayPal
Campus Party, São Paulo, Brasil, Jan 2014

Quem somos

The image displays two screenshots of payment service websites. On the left, the BillMeLater website is shown, featuring its logo ('BillMeLater® a PayPal® service') and a large blue navigation bar with tabs for 'Account', 'Store Directory', and 'About'. Below the navigation bar is a large image of a person wearing a blue and white patterned hat. On the right, the PayPal homepage is displayed, with the 'PayPal™' logo at the top, followed by links for 'Shopping', 'Offer Directory', 'Store Directory', and a search bar. A banner below the navigation bar reads 'Shop with PayPal' and features a section titled 'Save Big on Big Game Gear' with a subtext 'Great deals on everything you need for your football party.' and a 'Shop now' button. In the background of the PayPal screenshot, a person is seen sitting on a couch watching a football game on a television.

BillMeLater
a **PayPal** service

Account Store Directory About

PayPal™ Shopping Offer Directory Store Directory

Shop with PayPal

Save Big on Big Game Gear

Great deals on everything you need for your football party.

Shop now

PayPal

Tecnologias que Usamos

- Java: Tomcat / JBoss / Spring Framework
- Groovy: Linguagem de scripting para Java
- Grails: Web framework inspirado pelo Rails

Ciclo Tradicional de Waterfall

- Requisitos > Design > Código > Teste > Prod
Analista de Negócios > Dev > QA > Operações
- Ciclos de release longos
- Requisitos mudando durante o ciclo, mas sem uma maneira de gerenciá-los

Problema #1: Falta de comunicação entre equipes

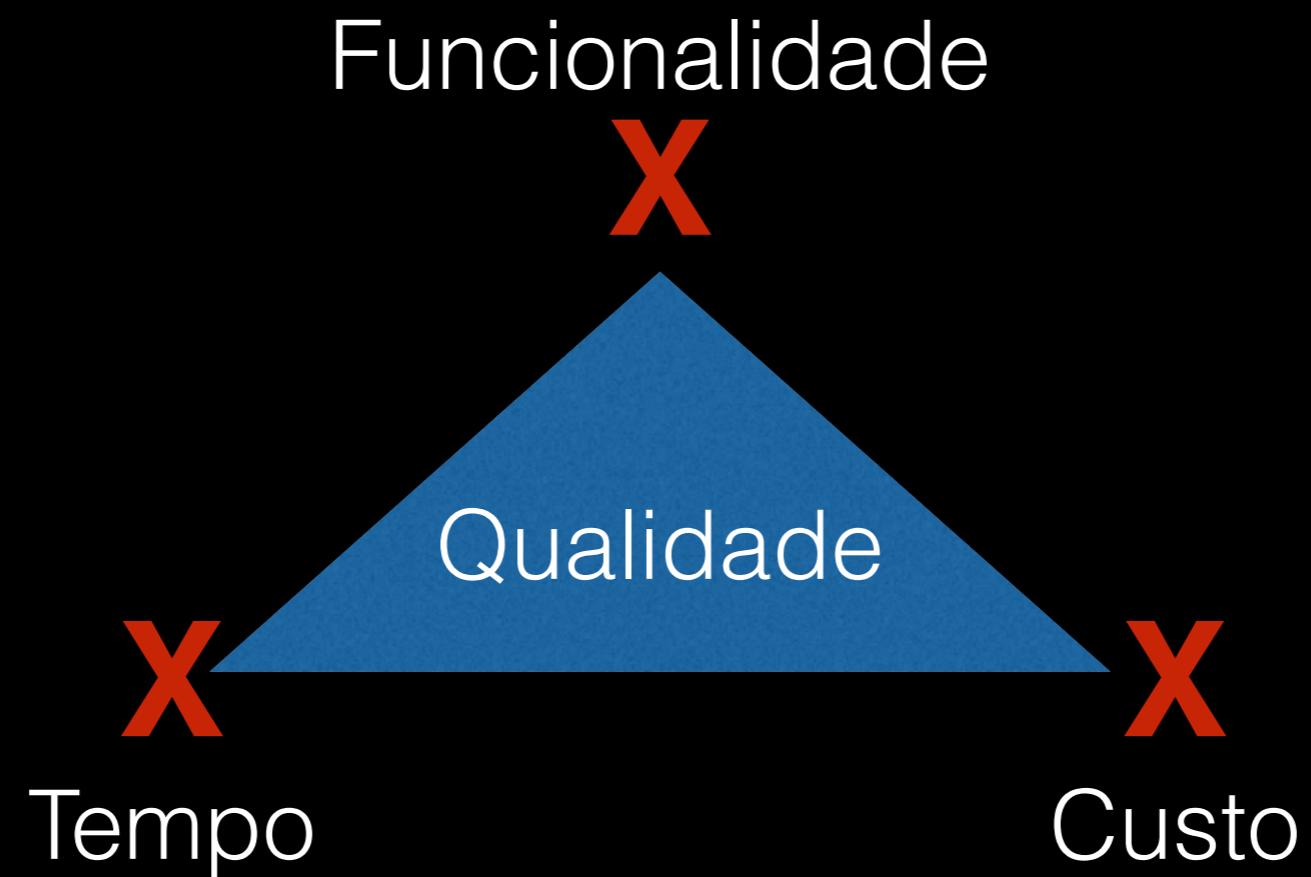
- Analistas de negócio vs. tecnologia
- Desenvolvimento vs. QA
- QA vs. Desenvolvimento

Problema #2: Ferramentas não promoviam colaboração

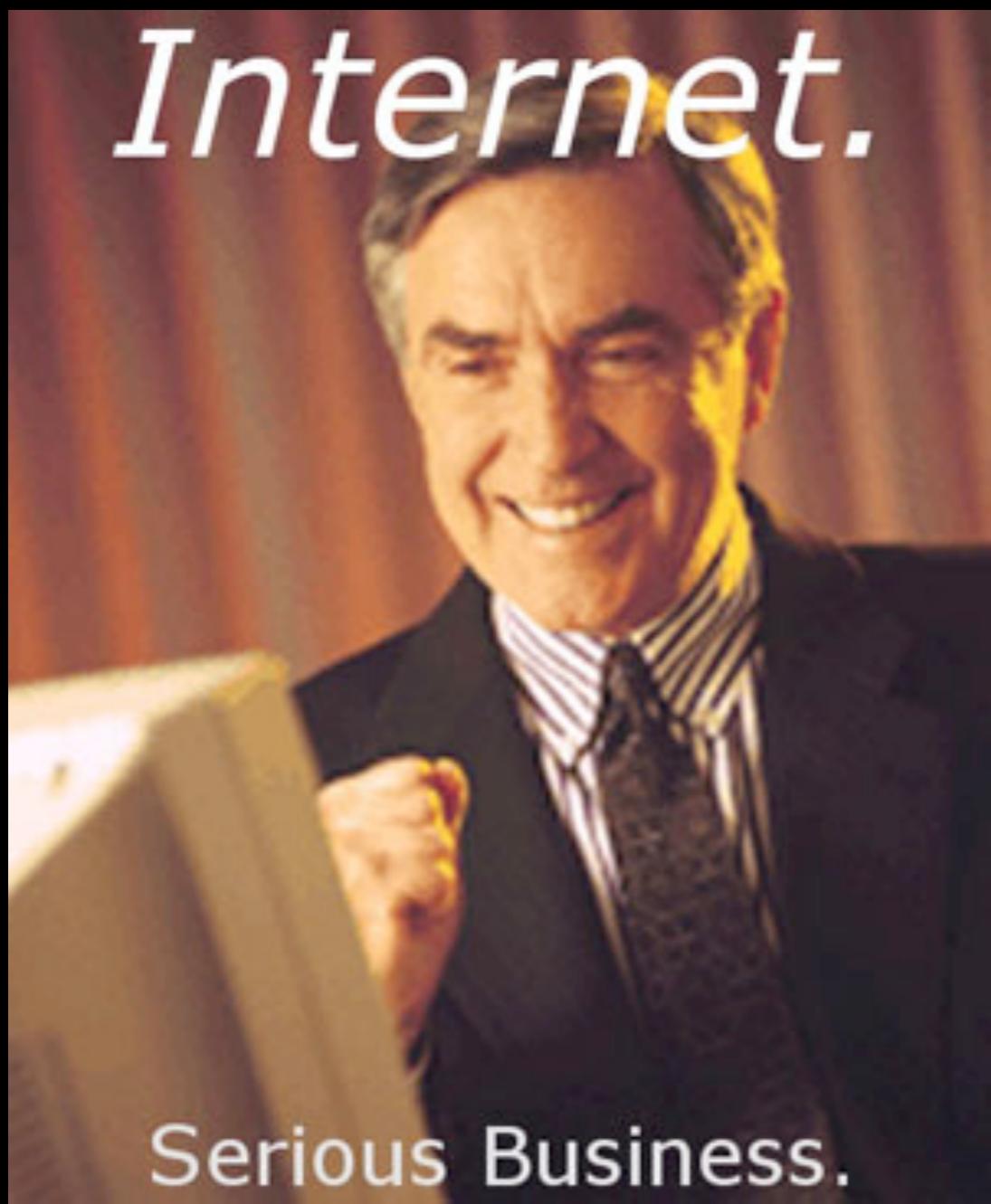
- Base de código de produção e testes mantidas separadamente.
- Dificuldade em manter testes atualizados com código de produção
- Resultado: Testes críticos executados manualmente

Problema #3:

Falta de processos para gerenciar mudanças no meio do release



Um pouco de teoria de engenharia de software



- **Analista de Negócio:**
Especifica requisitos
- **Desenvolvedor:**
Implementa especificação
- **QA:**
Verifica implementação
através da especificação

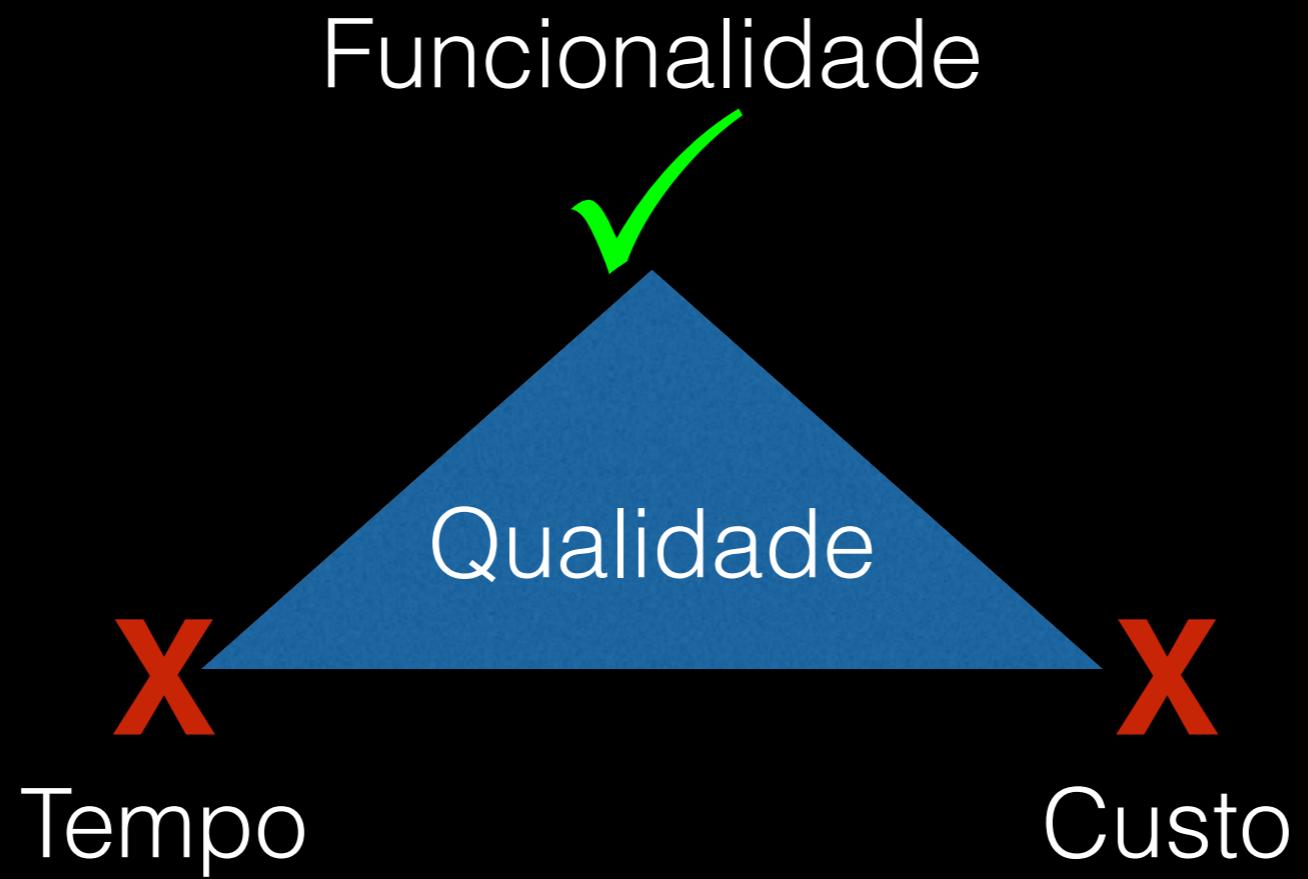
Transformação Agile na PayPal



Transformação Agile

Ciclos de 2 semanas:

1. Analista de Negócio define requisitos (user stories) e testes de aceitação;
2. Desenvolvedor implementa o código;
3. QA implementa testes automáticos baseados nos testes de aceitação;
4. Tudo roda em cima de Integração Contínua até que o release fique verde (= “user story está pronta”)



Ferramentas e Práticas

- Behavior-Driven Development (BDD)
- Automação de testes com Selenium e Geb
- Page Objects pattern

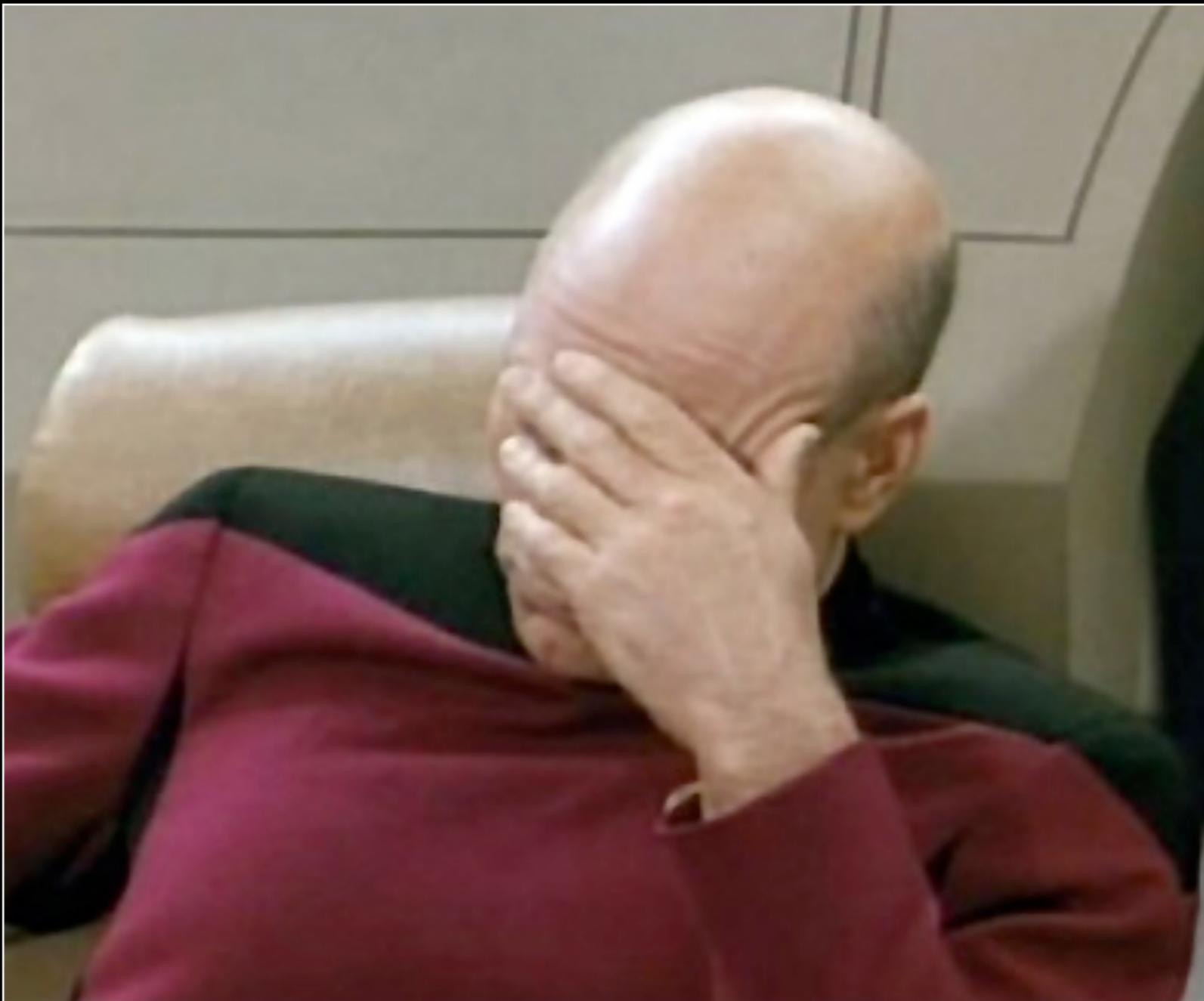
Selenium

- Framework para automação de testes de aplicações web
- Executa ações de usuário como clicar, teclar e verificar informações mostradas na página.
- Ações podem ser escritas em diversas linguagens de programação: Java, Ruby, C#, Python, etc...
- Suporta diversas combinações de browsers e OSs

Selenium Benefits

- Pode simular quase qualquer ação que um usuário pode executar.
- Testes são realistas, e muitos erros ignorados por outros métodos podem ser descobertos
 - Ex: Testes que dependam de configuração do browser, como teste com cookies ou JS desligados.
- API única facilita a criação de cross-browser testing

Demo Selenium



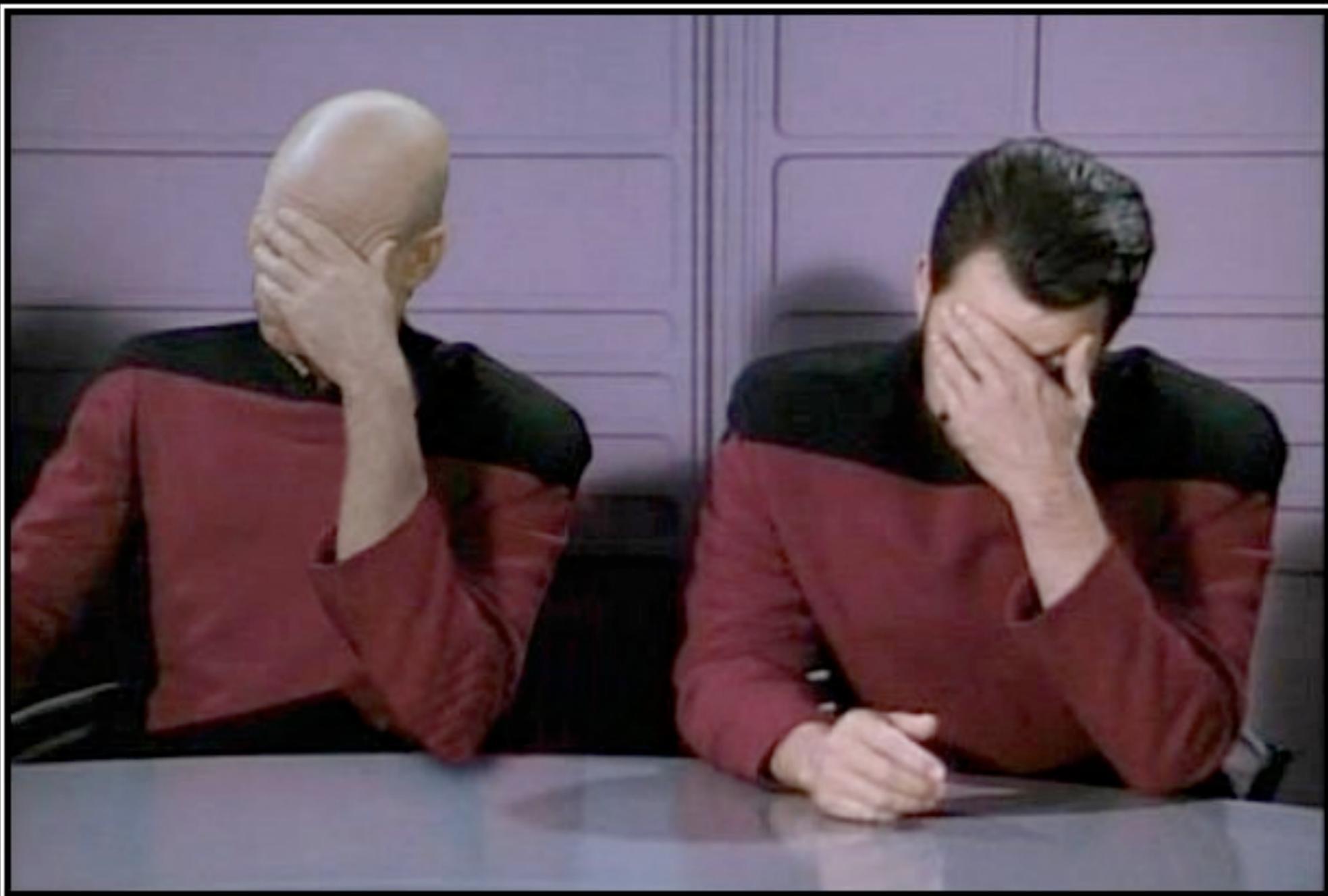
Geb e Groovy

- Geb: Framework baseado em Groovy que dá uma sintaxe parecida com o JQuery para os testes.
- Groovy: Linguagem de scripting para Java que simplifica tarefas e não requer conhecimentos de Java
- Groovy e Java co-existem no mesmo projeto:
 - Testes são mais legíveis e concisos em Groovy
 - Código Java pode ser chamado nativamente

Benefícios Geb e Groovy

- Ajuda a colaboração entre os times de UI e desenvolvimento:
 - Geb se parece com JQuery
 - Muito legível para equipes de UI.
- Fácil de aprender para testers que não são especialistas em Java
- It is an obvious choice if the dev team is Java-centric, but still very useful if dev team uses any other languages.

Demo Geb



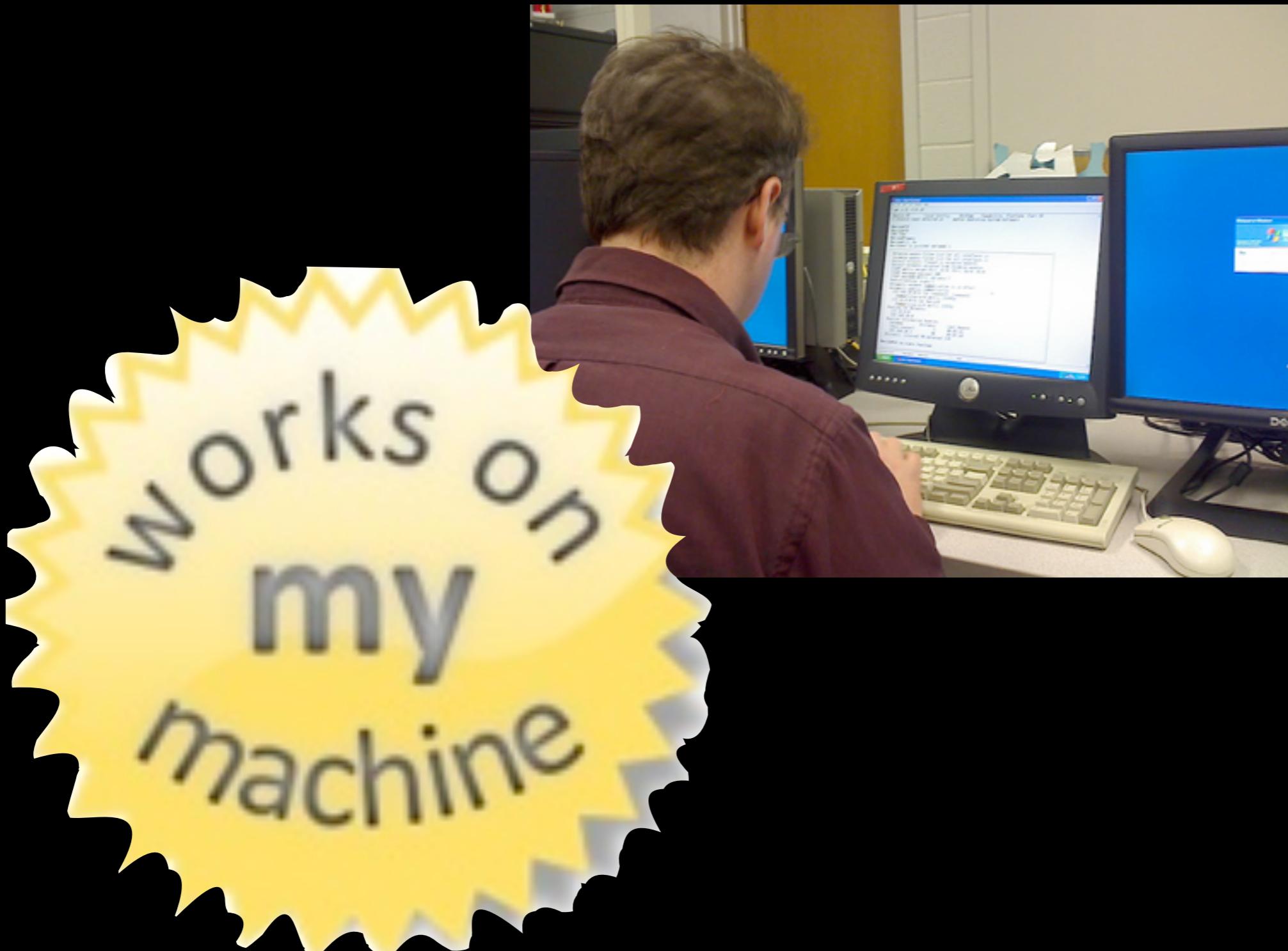
BDD

- Behavior-Driven Development (Cucumber)
- Usa linguagem de negócio para definir o comportamento do software e dos testes de aceitação
- A linguagem de negócio guia o código de teste
- O analista de negócio escreve os testes antes da funcionalidade existir: O teste vai falhar.
- O desenvolvedor implementa o código até que todos os testes passem: A funcionalidade está completa.

Benefícios do BDD

- BDD permite que o analista de negócio defina os testes usando termos e linguagem própria.
- Ajuda devs e QA a entender a linguagem de negócio e terminologia, e serve também como uma documentação do sistema.
- Transfere a responsabilidade de decisões de negócio para o analista de negócio, pois ele escreve os testes de aceitação.
- Testes passando servem como indicador de progresso para a user story.

Demo Cucumber



Page Object Pattern

- Pattern que espelha as funções de uma página web em forma de API.
- Separa a lógica de ações de testes da página da lógica do teste em si.
- Permite a reutilização de lógica comum em múltiplos testes
- Mudanças na UI são isoladas em apenas um objeto
- Aumenta enormemente a manutenção de testes em projetos de médio e grande porte

Agora tudo rodando

- (se compilar, funciona)

Resistências Humanas



“Nós nunca fizemos isso assim!”

- Geb não é Java. Geb é como JQuery
- Geb é fácil, aumenta produtividade da equipe de QA.
- Use métricas e crie objetivos de cobertura de código para convencer a equipe
 - Sonar

Resistências Humanas

“Nós não somos desenvolvedores web e não sabemos HTML/CSS!”

(o que é necessário para seletores em Geb)

- Treinamento: Crie guias simples de como achar seletores usando Firefox ou Chrome Dev Tools
- Cookbooks: Códigos de exemplo prontos para copiar e colar
- Venda a idéia de que QA terá mais tempo para testes exploratórios se testes comuns forem automatizados.
- Identifique “champions” na organização de QA para espalhar as novas práticas e ajudar os demais.

Organizational Resistance

“Nós já temos uma ferramenta de testes!”

- Tente integrar Geb+Cucumber com suite atual.
- Se suite existente for em Java, Geb pode facilmente chamar lógica já existente. Ex: set-up e tear-down de dados ou APIs já existentes.
- Selenium é um standard hoje, e é provável que a ferramenta atual já a utilize. Geb é apenas uma camada em cima do Selenium.

Limitações Técnicas

Cross-browser testing não é totalmente grátis

- Selenium Grid permite a orquestração de testes entre agentes rodando desktops virtuais em browsers e OSs diferentes.
- Problema: Testes podem falhar por comportamento diferente de cada browser (exemplo: tempo de espera de cliques)
- Sauce Labs é um grid na nuvem. Elimina a necessidade de manter um laboratório interno.
(Nota: Isto não é uma recomendação oficial!)

1 2 3 5 6

Resumo

- BDD permite que o analista de negócio defina requisitos e testes de aceitação, fazendo a ponte entre negócio e dev.
- Geb facilita a automação de testes, trazendo best-practices (manutenção, reusabilidade) para a equipe de QA.
- Como Geb é Groovy, e Groovy é Java, também facilita a colaboração entre QA e desenvolvedores.
- Desenvolvedores tem mais tempo para focar em funcionalidade ao invés de destrinchar requisitos ou dar suporte a esforços de teste.

Última opção



Código de Ética de Engenharia de Software da ACM

“Aprove software apenas se ele é seguro, atende às especificações, testes são aprovados, e não reduza a qualidade de vida, não reduza a privacidade ou traga prejuízos ambientais.

O efeito final do trabalho deve ser ao bem público.”

Obrigado!

PayPal

Links para esta apresentação, código fonte, ferramentas, e como me contactar:

<http://bit.ly/cp-2014>

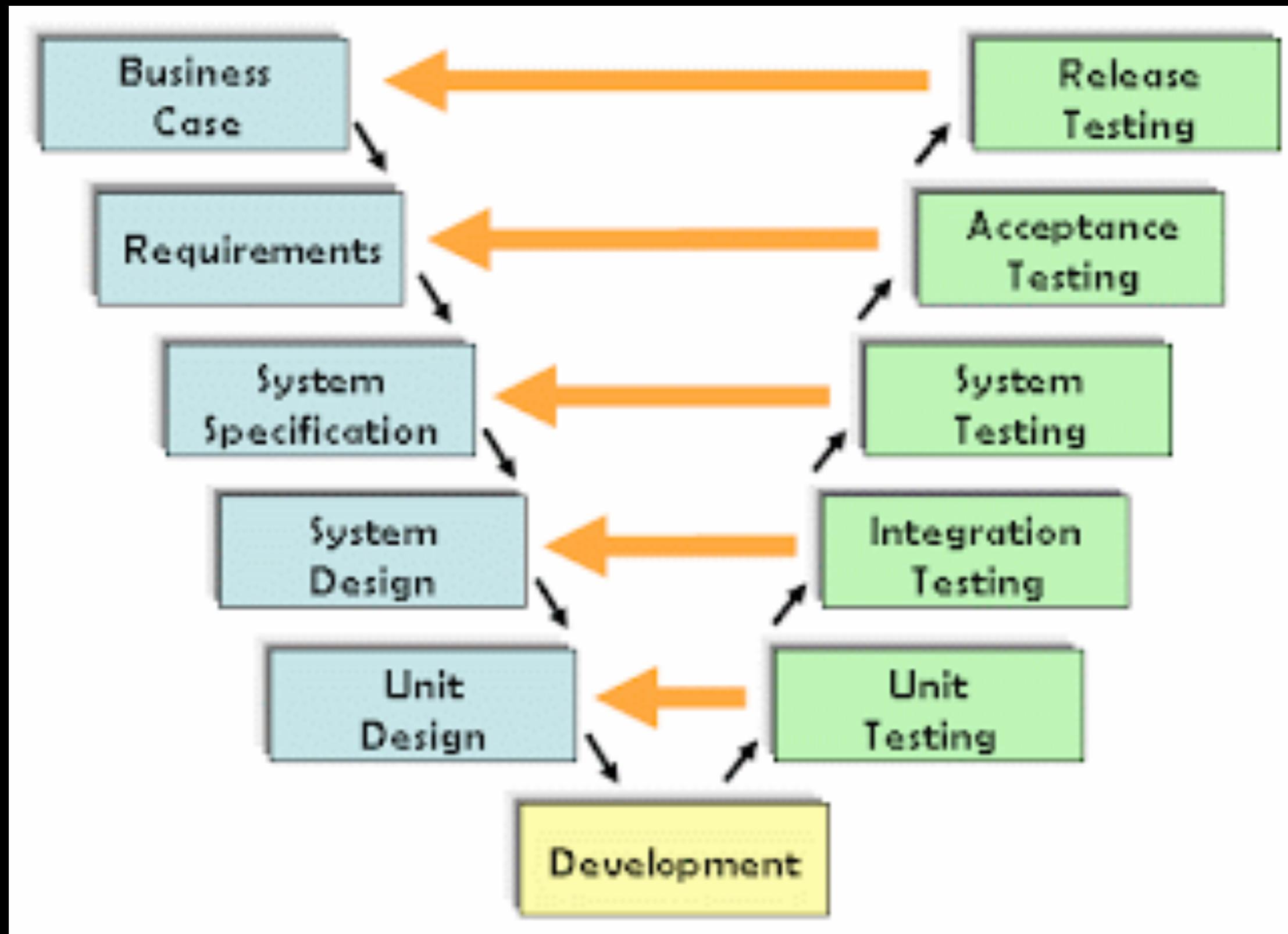
Images Usage Disclaimers (1)

- PayPal and BillMeLater screenshots (C) PayPal
- “Elevator Buttons”, Felipe Albertão
Creative Commons Attribution 2.0 Generic (CC BY 2.0)
- “Nerd Alert！”, William J Sisti
Creative Commons Attribution-ShareAlike 2.0 Generic (CC BY-SA 2.0) /
Superimposed with “Works on My Machine” pic
<http://www.flickr.com/photos/willsisti/411974911/>
- “Works on My Machine”, Duncan Hull
Creative Commons Attribution 2.0 Generic (CC BY 2.0) / Sub-imposed with
“Nerd Alert！” pic
<http://www.flickr.com/photos/dullhunk/7442633378/>
- “Left: You Are Doing It Wrong”, srslyguys
Creative Commons Attribution 2.0 Generic (CC BY 2.0)
<http://www.flickr.com/photos/srslyguys/2569202600/>
- “Picard Facepalm”, AJ Cann
Creative Commons Attribution-ShareAlike 2.0 Generic (CC BY-SA 2.0)
<http://www.flickr.com/photos/ajc1/6045169177/>

Images Usage Disclaimers (2)

- “Internet, Serious Business”, Clay Newton
Creative Commons Attribution-ShareAlike 2.0 Generic (CC BY-SA 2.0)
http://www.flickr.com/photos/sour_patch/1764038096/
- “Facepalm Picard Double”, darkuncle
Creative Commons Attribution 2.0 Generic (CC BY 2.0)
<http://www.flickr.com/photos/darkuncle/4421755040/>
- “Beer Night Aftermath”, Patrick Barry
Creative Commons Attribution-ShareAlike 2.0 Generic (CC BY-SA 2.0)
<http://www.flickr.com/photos/pbarry/5610452490/>
- “Dude Fest”, kelly blue
Creative Commons Attribution-NoDerivs 2.0 Generic (CC BY-ND 2.0)
<http://www.flickr.com/photos/kellyblue/3606708963/>

Back-up Slides



Developer



Requests IE
on Windows



selenium-hub

Delegates to IE



Selenium Grid

Connects to application under test (e.g. on developers machine)

BDD vs. TDD

- TDD (Test Driven Development)
 1. Developer writes test without any code: Test will fail (red)
 2. Developer writes code until test passes (green)
 3. Developer refactors code to remove replication
 4. Developer repeats red-green-refactor cycle until feature is complete
- BDD (Behavior Driven Development)
 5. Business Analyst writes test steps without any code: Test will fail (red)
 6. Developer writes code until test passes (green)
 7. Developer repeats red-green cycle until feature is complete
- BDD is a team practice. TDD is an individual developer practice.
- TDD can be used by the individual developer within the BDD practice.

