

[IF977] Engenharia de Software

Prof. Vinicius Cardoso Garcia
vcg@cin.ufpe.br :: [@vinicius3w](https://twitter.com/vinicio3w) :: assertlab.com

Continuação



Licença do material

Este Trabalho foi licenciado com uma Licença

**Creative Commons - Atribuição-NãoComercial-
Compartilhualgual 3.0 Não Adaptada.**

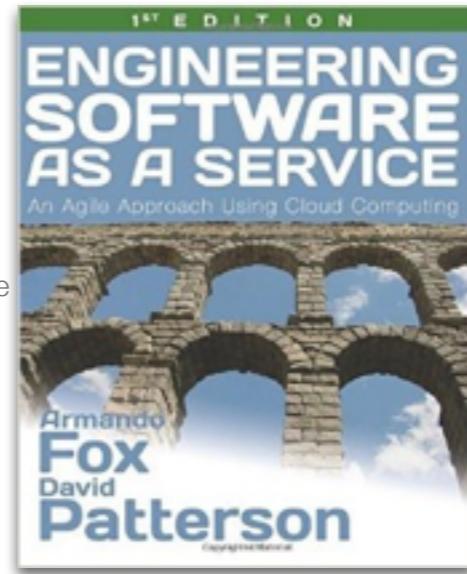
Mais informações visite

<http://creativecommons.org/licenses/by-nc-sa/3.0/deed.pt>



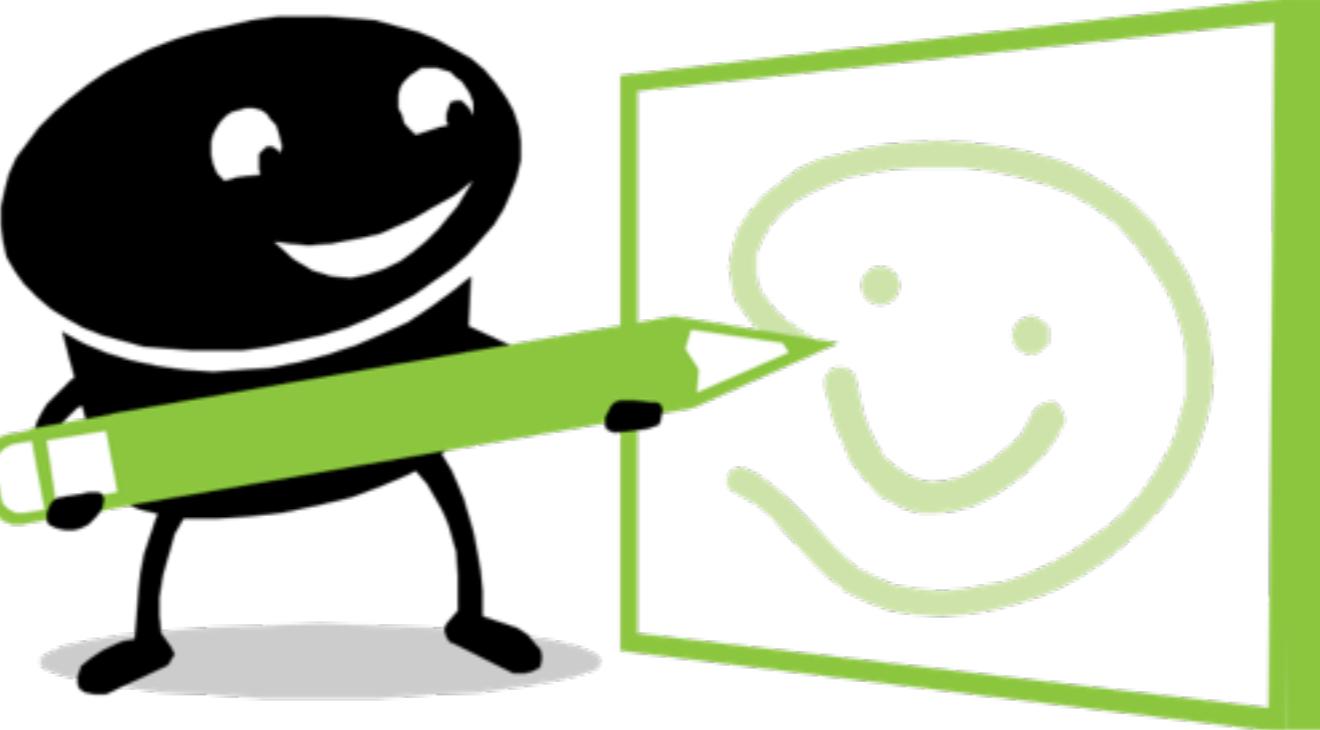
Referências

- A biblioteca do Desenvolvedor de Software dos dias de hoje
 - <http://bit.ly/TDOA5L>
- SWEBOK
 - Guide to the Software Engineering Body of Knowledge (SWEBOK): <http://www.computer.org/web/swebok>
- Engineering Software as a Service: An Agile Approach Using Cloud Computing (Beta Edition)
 - <http://www.saasbook.info/>



Outline

- ~~(ESaaS 9.5) Identifying What's Wrong: Smells, Metrics, SOFA~~
- ~~(ESaaS 7.1) Intro to BDD & User Stories~~
- Administrivia
 - ~~(ESaaS 7.3) Points, Velocity, and Pivotal Tracker~~
 - ~~(ESaaS 7.2) SMART User Stories~~
 - ~~(ESaaS 7.7) Lo-Fi User Interface Sketches and Storyboards~~
 - ~~(ESaaS 7.4) Agile Cost Estimation~~
 - ~~(ESaaS 7.9) Explicit vs. Implicit and Imperative vs. Declarative Scenarios~~
 - ~~(ESaaS 7.10-7.12) Fallacies & Pitfalls, BDD Pros & Cons~~
 - ~~(ESaaS 7.10) Plan-and-Document Perspective~~



Lo-Fi UI Sketches and Storyboards

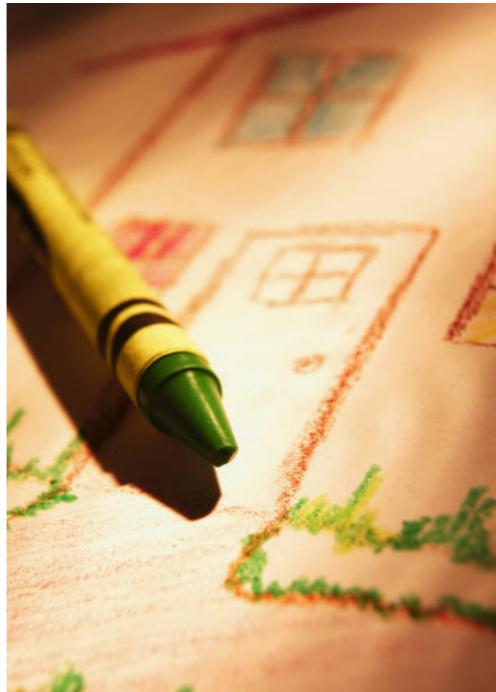
Construindo UI com sucesso

- Apps SaaS vezas atrapalham usuários
- As histórias de usuário precisam de Interfaces de Usuário (UI - User Interface)
- Como conseguir clientes para participar de projetos de UI que se sintam felizes quando finalizado o projeto?
 - Evite WISBNWIW* em UI?
 - Versão UI de cartões 3x5?
- Como mostrar a interatividade sem a construção de protótipo?

* What-I-Said-But-Not-What-I-Want

Projeto de UI em SaaS

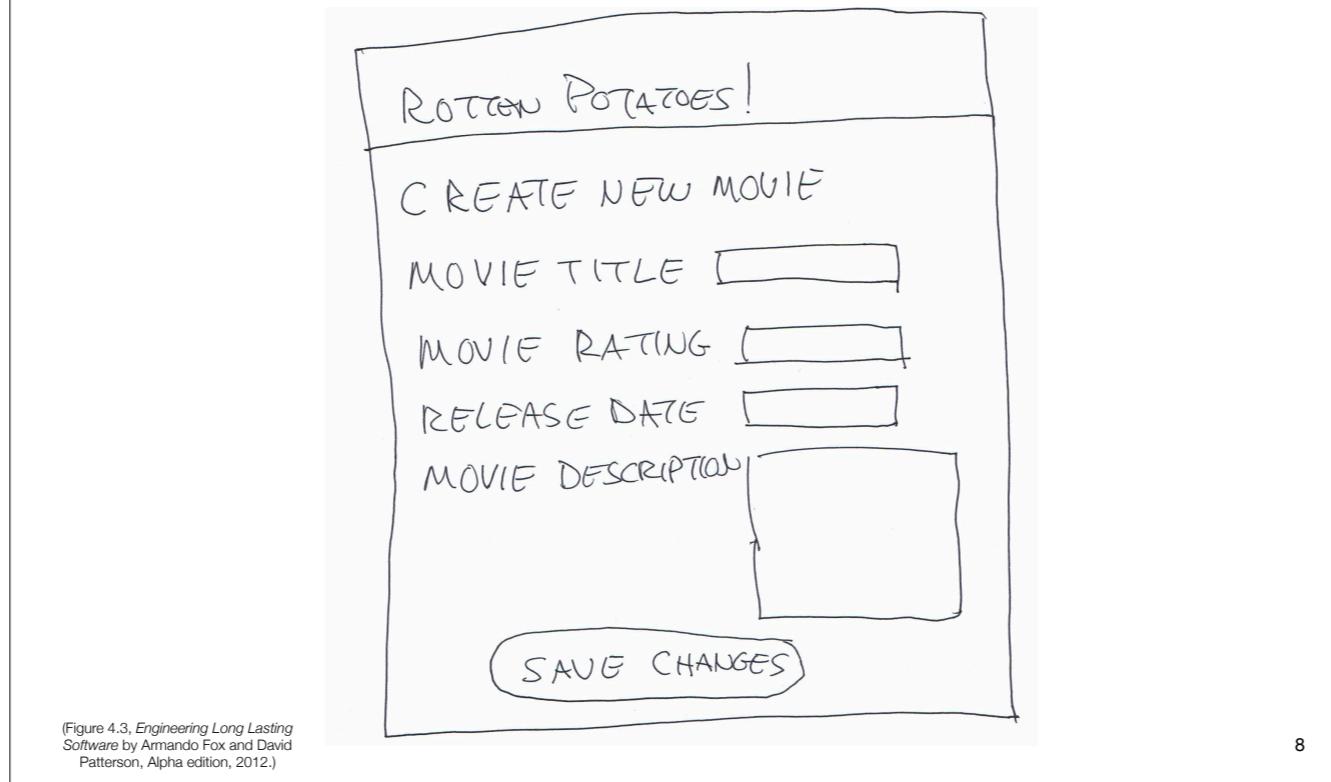
- **Sketches UI:** lápis e papel para desenhar ou “**Lo-Fi UI**”



7

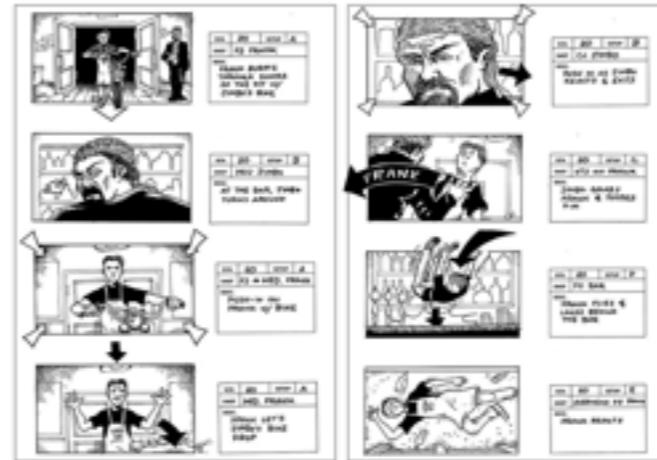
Lo-Fi: low fidelity

Exemplo de Lo-Fi UI



Storyboards

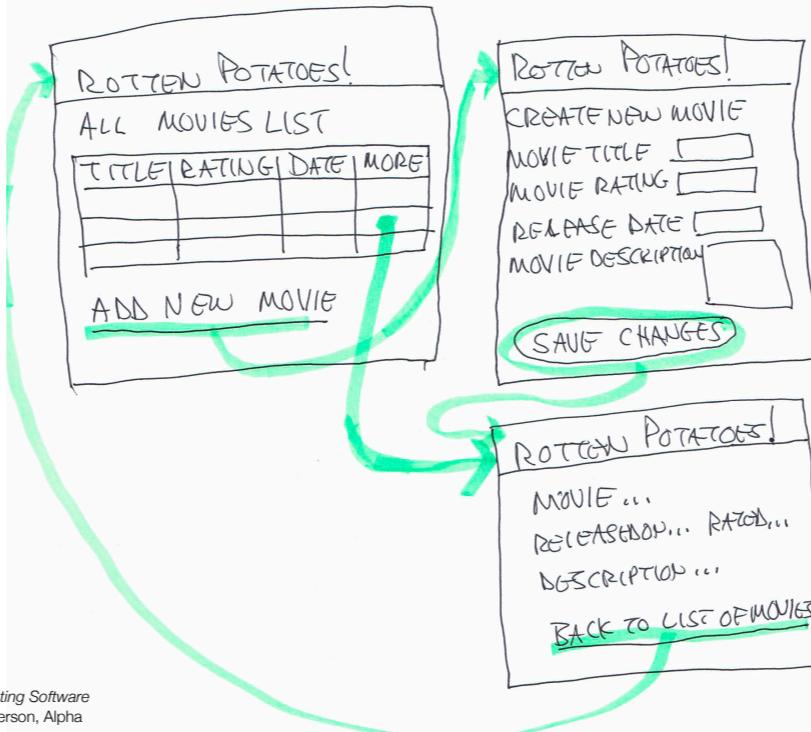
- Necessitam mostrar como a UI muda a partir de ações dos usuários
- IHC (HCI) => “storyboards”
 - Como cenas em um filme
 - Mas sem ser “linear”



9

HCI, Human-Computer Interaction

Exemplo de Storyboard para Software



(Figure 4.4, *Engineering Long Lasting Software*
by Armando Fox and David Patterson, Alpha
edition, 2012.)

Lo-Fi para HTML

- É entediante construir sketches & storyboards, porém mais fácil do que construir HTML!
 - Além disso, **menos intimidador** para stakeholders **não-técnicos** => **incentivador** à sugestão de **modificações** de UI se não houver **código** por trás
 - Uma maior probabilidade de sucesso com a UI final
 - Próximos passos: CSS (Cascading Style Sheets) & Haml
 - Melhorar a aparência depois de funcionando

Pergunta

O que é **FALSO** sobre Lo-Fi UI?

- A. Assim como cartões 3x5, sketches & storyboards são mais propensos a envolver todos os stakeholders vs. código
- B. O propósito da abordagem Lo-Fi UI é depurar a UI antes de você programá-la
- C. Apps SaaS geralmente têm UIs associadas às suas histórias do usuário
- D. Enquanto leva mais tempo do que a construção de um protótipo de UI em CSS e Haml, a abordagem Lo-Fi é mais susceptível de conduzir a uma interface de usuário que os clientes gostam

Pergunta

O que é **FALSO** sobre Lo-Fi UI?

- A. Assim como cartões 3x5, sketches & storyboards são mais propensos a envolver todos os stakeholders vs. código
- B. O propósito da abordagem Lo-Fi UI é depurar a UI antes de você programá-la
- C. Apps SaaS geralmente têm UIs associadas às suas histórias do usuário
- D. Enquanto leva mais tempo do que a construção de um protótipo de UI em CSS e Haml, a abordagem Lo-Fi é mais susceptível de conduzir a uma interface de usuário que os clientes gostam





Estimativa de Custo Ágil

(Engineering Software as a Service §7.4)

Estimativa de Custo Ágil

- O mundo real precisa estimar o custo antes de assinar qualquer termo de compromisso de um projeto
- Se não houver um planejamento nas Metodologias Ágeis, como eu posso estimar o custo de um projeto?

Modelo da Pivotal Labs

- Pivotal Labs ensina Metodologias Ágeis aos clientes
- Utilizando Metodologias Ágeis, a Pivotal **nunca** se compromete a entregar as features X, Y, and Z até a data D
- Ao invés disso, são comprometidos **recursos** para trabalhar da forma mais eficiente possível até a data D
 - Clientes trabalham **com o time** para definir as prioridades continuamente até a data D
 - Ainda assim, necessitamos de estimativas para o projeto

Estimativa Ágil da Pivotal Labs

1. Telefonema de ~1 hora para explicar o método
 - Esforço conjunto, compromisso do cliente, ...
2. Visitas do cliente de ~1.5 horas para definição/entendimento do escopo
 - Clientes leva designer, desenvolvedor, projetos
 - Qualquer coisa para clarificar o que precisa estar pronto
 - Pivotal leva 2 engenheiros para fazer perguntas
 - Identificar o que adiciona incerteza à estimativa
3. Engenheiros tomam ½ hora para estimar as semanas
 - Pequena vs. grande incerteza: 20-22 vs. 18-26 semanas
4. Custo da oferta como tempo e material para o cliente

Pergunta

Qual expressão sobre estimativa de custo é **VERDADEIRA**?
(PL = Pivotal Labs)

- A. Como praticantes do Desenvolvimento Ágil, PL não utiliza contratos
- B. Como praticantes da programação em pares, PL estima o custo para um par, ao qual ela atribui a responsabilidade de completar o projeto
- C. A oferta de custo para PL é por tempo e materiais que abrangem um número de semanas na estimativa
- D. Como alguns estudos mostram, 84% a 90% dos projetos são executados no prazo e dentro do orçamento, os gestores Dirigidos a Plano prometem aos clientes um conjunto de funcionalidades para um custo acordado para uma data estabelecida

18

1. Pivotal certainly offers customers a contract that they sign, but it is primarily a promise to pay Pivotal for its best effort over to make the customer happy for a limited range of time.
2. Close. They will put multiple pairs on the project, the number depending on what is most efficient size team.
3. True
4. False. – 84% to 90% project are late and/or over budget

Pergunta

Qual expressão sobre estimativa de custo é **VERDADEIRA**?
(PL = Pivotal Labs)

- A. Como praticantes do Desenvolvimento Ágil, PL não utiliza contratos
- B. Como praticantes da programação em pares, PL estima o custo para um par, ao qual ela atribui a responsabilidade de completar o projeto
-  C. A oferta de custo para PL é por tempo e materiais que abrangem um número de semanas na estimativa
- D. Como alguns estudos mostram, 84% a 90% dos projetos são executados no prazo e dentro do orçamento, os gestores Dirigidos a Plano prometem aos clientes um conjunto de funcionalidades para um custo acordado para uma data estabelecida

19

1. Pivotal certainly offers customers a contract that they sign, but it is primarily a promise to pay Pivotal for its best effort over to make the customer happy for a limited range of time.
2. Close. They will put multiple pairs on the project, the number depending on what is most efficient size team.
3. True
4. False. – 84% to 90% project are late and/or over budget



Cenários Explícitos vs.
Implícitos e Imperativos vs.
Declarativos

Explícito vs Implícito

- Requisitos explícitos geralmente partem de testes de aceitação
 - Como histórias e cenários de usuários explícitas: Listar filmes
- Requisitos implícitos são consequências lógicas dos requisitos explícitos, tipicamente testes de integração
 - Filmes listados em ordem cronológica ou ordem alfabética?

Imperativos vs. Declarativos

- Imperativo: histórias de usuários iniciais com muitos passos, especificando sequências lógicas para os resultados desejados
 - Not-DRY se existirem muitas histórias de usuários imperativas
 - Declarativa: descrever estado, não a sequência
 - Menos passos
 - Feature: filmes devem aparecer em ordem alfabética, não ordem de registro
 - Cenário: lista de filmes vista após a adição de 2 filmes

Exemplo de Cenário Imperativo

```
Given I am on the Create New Movie page
  RottenPotatoes home page
When I follow "Add new When I fill in "Title" with
  movie" "Apocalypse Now"
Then I should be on the And I select "R" from
  Create New Movie page "Rating"
When I fill in "Title" with And I press "Save Changes"
  "Zorro" Then I should be on the
And I select "PG" from RottenPotatoes home page
  "Rating"
And I press "Save Changes" When I follow "Movie Title"
Then I should be on the Then I should see
  RottenPotatoes home page "Apocalypse Now" before
When I follow "Add new "Zorro"
  movie"
Then I should be on the
```

Apenas um passo especificando
comportamento;
O resto é implementação. Mas BDD
especifica o comportamento, não a
implementação!

Linguagem de Domínio

- Declarativa como se fosse uma linguagem de domínio
- Usa termos e conceitos de aplicativo
- Linguagem informal
- Passos declarativas descrevem o estado em que a app deve estar
 - Imperativo: sequência de passos para alterar do estado atual para o estado desejado

Exemplo de Cenário Declarativo

```
Feature: movies when added  
    should appear in movie  
    list
```

```
Scenario: view movie list  
    after adding movie  
    (declarative and DRY)
```

```
Given I have added  
    "Zorro" with rating "PG-  
    13"
```

```
And I have added  
    "Apocalypse Now" with  
    rating "R"
```

```
Then I should see  
    "Apocalypse Now" before  
    "Zorro" on the Rotten
```

Potatoes home page

**3 passos vs. 15 passos:
2 para configurar o teste,
1 para comportamento**

**Cenários declarativos focam
atenção na feature que está
sendo descrita e testada vs
passos necessários para
configurar o teste**

E novas definições de passos?

Cenários Declarativos necessitam de novas Definições de Passos

```
Given /I have added "(.*)" Then /I should see "(.*)"
  with rating "(.*)"/ do      before "(.*)" on (.*)/ do
    |title, rating|           |string1, string2, path|
  Given I am on the Create    step "I am on #{path}"
  New Movie page             regexp =
  When I fill in "Title"     /#{string1}.*#{string2}/m
  with "#{title}"            # /m means match across
  And I select "#{rating}"   newlines
  from "Rating"              page.body.should =~
  And I press "Save"          regexp
  Changes"                  end
end
```

- À medida que o app evolui, reutilize os passos dos primeiros cenários imperativos para criar cenários declarativos mais concisos e descritivos

Pergunta

O que é **VERDADE** sobre cenários implícitos vs. explícitos e imperativos vs. declarativas?

- A. Requisitos explícitos são geralmente definidos com cenários imperativas e requisitos implícitos são geralmente definidos com cenários declarativos
- B. Cenários explícitos geralmente capturam testes de integração
- C. Cenários declarativos ajudam a capturar implementações assim como comportamentos
- D. Todas são falsas

27

1st false: These are two independent classifications; both requirements can use either type of scenarios

2nd false: Explicit stories usually acceptance tests

3rd false: Declarative aiming at behavior

4th All are false.

Pergunta

O que é **VERDADE** sobre cenários implícitos vs. explícitos e imperativos vs. declarativas?

- A. Requisitos explícitos são geralmente definidos com cenários imperativas e requisitos implícitos são geralmente definidos com cenários declarativos
- B. Cenários explícitos geralmente capturam testes de integração
- C. Cenários declarativos ajudam a capturar implementações assim como comportamentos
- D. Todas são falsas



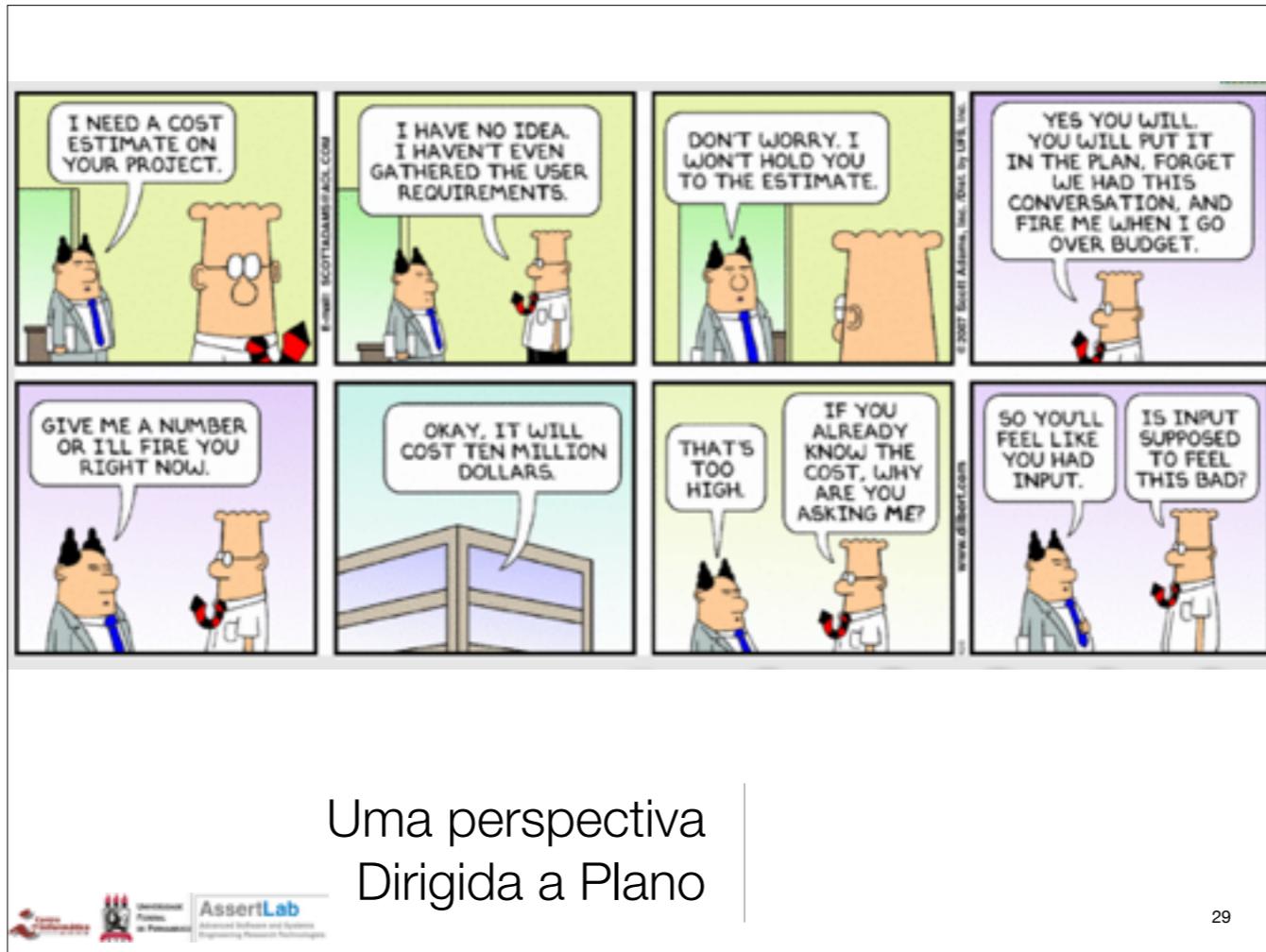
28

1st false: These are two independent classifications; both requirements can use either type of scenarios

2nd false: Explicit stories usually acceptance tests

3rd false: Declarative aiming at behavior

4th All are false.



Uma perspectiva Dirigida a Plano

Introdução

- O que abordagens Dirigidas a Plano fazem ao invés de
 - Histórias do Usuário?
 - Pontos?
 - Velocidade?
- Como um gerente de projeto estima os custos? faz um cronograma?



P&D Equivalentes+

- Elicitação de Requisitos
- Documentação de Requisitos
- Estimativa de Custo
- Cronogramas e Monitoramento do Progresso
- Gerenciamento de Mudanças para Requisitos, Custo & Cronograma
- Garantir que a Implementação esteja de acordo com os Requisitos
- Análise e Gerenciamento de Riscos



Elicitação de Requisitos :: P&D

1. Elicitar requisitos **funcionais** & **não-funcionais**:

- **Entrevistas** - veja como é atualmente feito
 - Stakeholders respondem perguntas
 - Ou apenas tenha conversas informais

2. Crie **cenários** cooperativamente

- Estado inicial, apresente fluxos para caminhos felizes e tristes, qual é o estado atual e o final

3. Crie **Casos de Uso**

- Lista de passos para alcançar um objetivo entre o usuário e o sistema; existe uma linguagem (UML) para descrever de maneira abstrata

32

Functional – features, “add a movie”

Non-functional – performance goal (eg/. 99% responses) < 1 sec or dependability goal (4 9s)

Documentação de Requisitos :: P&D

- Documentar requisitos através de uma **Especificação de Requisitos de Software (SRS)**
 - 100s de páginas; IEEE standard para SRS!
 - Os stakeholders leram o SRS, ou construiu um protótipo básico, ou gerou os casos de teste para verificar:
 - **Validade:** todos os requisitos são necessários?
 - **Consistência:** existem conflitos nos requisitos?
 - **Completude:** todos os requisitos e restrições foram incluídas?
 - **Viabilidade:** os requisitos são exequíveis?

33

SRS, Software Requirements Specification

830-1998 — IEEE Recommended Practice for Software Requirements Specifications. 1998. doi:10.1109/IEEESTD.1998.88286. ISBN 0-7381-0332-2. URL: <http://dx.doi.org/10.1109/IEEESTD.1998.88286>

Estimativa de Custo :: P&D

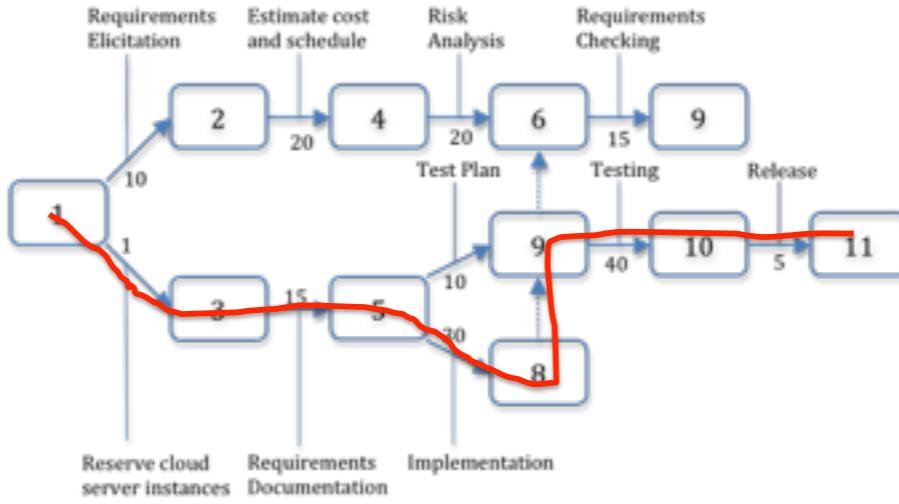
- Gerente分解 o SRS em tarefas
- Estima semanas por tarefas
 - 1 semana <= Tarefas <= 8 semanas
- Converta pessoas/semana em \$ via salários e despesas gerais
- Estime antes e depois do contrato
 - Adicionar uma margem de segurança: 1,3 a 1,5X
 - Faça 3 estimativas (melhor, esperado, pior caso), em seguida executar o melhor palpite

Estimativa de Custo :: P&D

- Estimativa Experimental/Empírica
 - Confie na experiência do gestor para ser exato
- Estimativa quantitativa
 - Estimar tarefas em linhas de código (LOC), dívida por LOC/pessoa-mês
 - COCOMO (Constructive Cost Model):
$$\text{Effort} = \text{OrgFactor} * \text{CodeSize}^{\text{Penalty}} * \text{ProdFactor}$$
 - Factor de Organização = 2,94, $1.10 \leq \text{SizePenalty} \leq 1.24$, Fator 0,90 ≤ Produto ≤ 1,40
 - 92% usam a experimental vs. a fórmula

Scheduling :: P&D

- Uso de redes de PERT para exibir paralelismo nas tarefas e caminhos críticos para auxiliar na programação



- Nós são marcos (milestones), nomes de link são tarefas, números são métrica de esforço, setas são dependências

Monitorando o Progresso :: P&D

- Comparar o previsto para real
 - Tempo para as tarefas
 - Despesas
- Marcos intermediários ajudam todos os stakeholders a verificar se o projeto está no **cronograma** e no **orçamento**



Validação dos Requisitos :: P&D

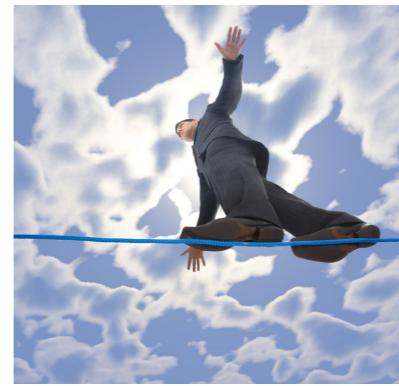
- Gerentes utilizam ferramentas para **rastreabilidade dos requisitos**
- Ferramentas possuem referências cruzadas entre elas
 - Parcada do SRS com requisitos
 - Parte do código que implementa o requisito
 - Testes que validam o requisito



38

Análise e Gerenciamento de Riscos :: P&D

- Melhorar a acurácia de investimento/cronograma
 - Identificar riscos mais cedo para
 - Trabalhar na redução dos riscos
 - Mudar o plano para evitar os riscos
- Técnico: RDB não escala
- Organizacional: J2EE???
- Negócio: muito tarde para o mercado
- Crie uma Tabela de Riscos: % probabilidade x impacto
- Ataque os 20% principais, esperança para os 80%



Requisitos e Estimativa de Custo/Cronograma Ágil vs P&D

<i>Tasks</i>	<i>In Plan and Document</i>	<i>In Agile</i>
Requirements Documentation	Software Requirements Specification such as IEEE Standard 830-1998	User stories, Cucumber, Points, Velocity
Requirements Elicitation	Interviews, Scenarios, Use Cases	
Change Management for Requirements, Schedule, and Budget	Version Control for Documentation and Code	
Ensuring Requirements Features	Traceability to link features to tests, reviews, and code	
Scheduling and Monitoring	Early in project, contracted delivery date based on cost estimation, using PERT charts. Milestones to monitor progress	
Cost Estimation	Early in project, contracted cost based on manager experience or estimates of task size combined with productivity metrics	Evaluate to pick range of effort for time and materials contract
Risk Management	Early in project, identify risks to budget and schedule, and take actions to overcome or avoid them	

Figure 7.14: The relationship between the requirements related tasks of Plan-and-Document versus Agile methodologies.

Pergunta

Qual expressão relacionada a Estimativa de Custo e Cronograma em P&D é **FALSA**?

- A. O mais próximo às tarefas de planejamento e monitoramento de tarefas do P&D são pontos e velocidade no Desenvolvimento Ágil
- B. O mais próximo ao documento de Especificação de Requisitos de Software (SRS) no Desenvolvimento Ágil são as histórias do usuário
- C. Desenvolvimento Ágil não tem equivalente no que diz respeito a garantia de requisitos, como a rastreabilidade
- D. Atualmente, 1, 2, e 3 são verdadeiras; nenhuma é falsa

41

1. True.
2. True.
3. False. Agile ensures requirements via Behavior Driven Design by turning user stories into acceptance tests
4. True.

Pergunta

Qual expressão relacionada a Estimativa de Custo e Cronograma em P&D é **FALSA**?

- A. O mais próximo às tarefas de planejamento e monitoramento de tarefas do P&D são pontos e velocidade no Desenvolvimento Ágil
- B. O mais próximo ao documento de Especificação de Requisitos de Software (SRS) no Desenvolvimento Ágil são as histórias do usuário
-  C. Desenvolvimento Ágil não tem equivalente no que diz respeito a garantia de requisitos, como a rastreabilidade
- D. Atualmente, 1, 2, e 3 são verdadeiras; nenhuma é falsa

Concluindo...

- Métodos CUPA: Curto, faça Uma coisa,
Poucos argumentos, Abstração consistente
 - Métricas apontam para partes possivelmente problemáticas do programa
- BDD: Histórias do usuário para eliciar requisitos
 - SMART: *Specific, Measureable, Achievable, Relevant, Timeboxed*
- Pontos/Velocidade para calcular o progresso (Tracker)
- Lo-Fi UI/storyboard: sketch para eliciar projeto de UI
- Ciclo de vida: Dirigido a Plano (planejamento e documentação cuidadosa) v.
Ágeis (abrace a mudança)

43

(Engineering Software as a Service §9.4-9.5, 7.1-7.4, 7.7)



Falácia e Armadilhas, BDD Pros & Cons

(Engineering Software as a Service §7.10-§7.12)

Armadilha

- Clientes confundem mockups digitais com features concluídas
 - Clientes não técnicos acham altamente polidas mockup digitais = features funcionais
- Use mockups Lo-Fi, como representações claras de recursos propostos

Armadilha

- Sketches sem storyboards
 - Precisam chegar a um acordo com o cliente na interação com páginas, bem como com o conteúdo da página
 - Storyboards / sketches "animados" reduzem mal-entendidos

Armadilha

- Adicionar recursos interessantes que não tornam o produto mais bem sucedido
 - Clientes rejeitam o que os programadores gostam!
- Tentar prever o código que você precisa antes de precisar dele
 - BDD: escreva testes antes de escrever o código que você precisa, em seguida, escreva o código necessário para passar nos testes
- Histórias de usuários ajudam a priorizar & BDD minimiza o que você codifica => reduzir o desperdício de esforço

Armadilha

- Entregar uma história como pronta apenas quando o caminho feliz é testado
 - Precisamos testar tanto caminho feliz quanto os tristes
 - Corrigir o comportamento da app quando o usuário acidentalmente faz alguma coisa errada é tão importante quanto comportamento correto quando ele faz coisa certa
 - Errar é humano

Armadilha

- O uso descuidado de expectativas negativas
 - Cuidado com o uso excessivo "Então eu não deveria ver ...".
 - Não posso dizer se a saída é o que queremos, mas apenas que não é o que você quer
 - Muitas, muitas saídas estão incorretas
- Incluir positivos para verificar os resultados
 - "Então eu deveria ver ..."

Armadilha

- O uso descuidado de expectativas positivas
 - **Then I should see "Emma"**
 - e se a **string** aparece várias vezes na página?
 - Pode passar até mesmo se o recurso não funciona
 - Use o ajudante **within** do Capybara
 - Restringe o escopo da comparação em um seletor CSS
 - **Then I should see "Emma" within "div#shopping_cart"**
 - Consulte a documentação Capybara

Pergunta

Qual expressão sobre Lo-Fi UI & BDD é **FALSA**?

- A. O objetivo da abordagem Lo-Fi UI é para depurar a interface do usuário antes de programá-la
- B. Uma desvantagem de BDD é exigir um contato permanente com os clientes, o que pode não ser possível
- C. Uma desvantagem de BDD é que ele pode conduzir a uma arquitetura de software pobre, uma vez que o foco é no comportamento
- D. Nenhuma delas é falsas; as três acima são verdadeiras

4th one; all true

Pergunta

Qual expressão sobre Lo-Fi UI & BDD é **FALSA**?

- A. O objetivo da abordagem Lo-Fi UI é para depurar a interface do usuário antes de programá-la
- B. Uma desvantagem de BDD é exigir um contato permanente com os clientes, o que pode não ser possível
- C. Uma desvantagem de BDD é que ele pode conduzir a uma arquitetura de software pobre, uma vez que o foco é no comportamento
-  D. Nenhuma delas é falsas; as três acima são verdadeiras

4th one; all true