

WORKSHOP

Produzindo Especificações Ágeis e Testes Funcionais com Concordia

THIAGO DELGADO PINTO

SEMANA DE EXTENSÃO 2019

BSI - CEFET/RJ NOVA FRIBURGO

13/05/2019



Licença Creative Commons 4



github.com/thiagodp



Professor do CEFET/RJ Nova Friburgo, desde 2009



Doutor em Computação, PUC-Rio, 2018

Mestre em Computação, PUC-Rio, 2013



Pós-graduado em Engenharia de Software, Universidade Senac Rio, 2009



Bacharel em Informática, UNESA Nova Friburgo, 2003



WAIS Tecnologia da Informação, 2002-2010



Thiago Delgado Pinto

Trabalha com testes?

Curso a disciplina "Teste e Manutenção de Software" ou similar?



motivações

DEFEITOS PROVENIENTES DE REQUISITOS INCOMPLETOS OU INCORRETOS SÃO EXTREMAMENTE CAROS

até
100x
mais caros



EMPRESAS ESTÃO BUSCANDO O DESENVOLVIMENTO ÁGIL

CASCATA



INCREMENTAL



ÁGIL



EMPRESAS ÁGEIS ESTÃO USANDO FUNCIONALIDADES E CENÁRIOS PARA ESPECIFICAR E VALIDAR REQUISITOS



<nome da funcionalidade>

Como um <papel>
Eu desejo <funcionalidade>
Para <benefício>



Funcionalidade: <nome>

Como um <papel>

Eu desejo <funcionalidade>

Para <benefício>

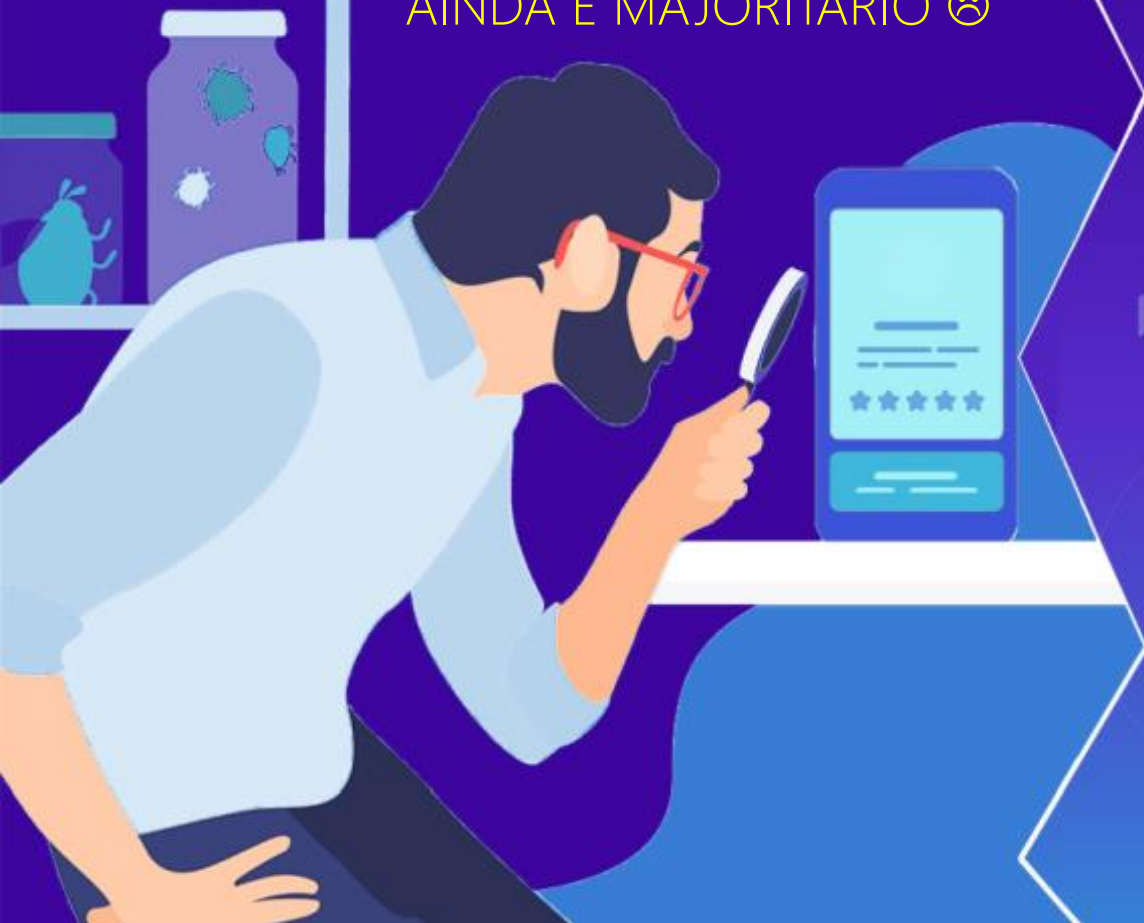
Cenário: <nome>

Dado que <pré-condição>

Quando <ação>

Então <resultado ou pós-cond.>

TESTE PURAMENTE MANUAL
AINDA É MAJORITÁRIO ☹



QUALIDADE DE TESTES NÃO GERADOS DEPENDEM DA HABILIDADE TÉCNICA DOS TESTADORES



COMO UNIR

Especificação Ágil de Requisitos

Redução de Defeitos em Requisitos

Geração Automática de Testes Completos

Garantia da Cobertura de Testes

Facilidade de Automação e Uso

?

Concordia

Especificação Ágil de Requisitos

Nova Metalíngua

Reuso de Conhecimento Existente

Separação de Língua de Negócios e de Tecnologia

Suporte a Múltiplos Idiomas

Sintaxe Simples e Flexível

Reconhecimento com Processamento de Língua Natural e I.A.

Redução de Defeitos em Requisitos

Verificação de Sintaxe, Semântica e Lógica

Sintaxe Simples para Facilitar Validação por Usuários/Interessados

Separação de Sintaxe de Tecnologia e de Negócios para Evitar Problemas de Comunicação

Geração Automática de Testes Completos

Mix de Novas Técnicas com Estado-da-Arte

Geração e Combinação de Casos de Teste

Integração com Arquivos e Bancos de Dados

Geração e Combinação de Dados de Teste

Geração de Oráculos

Geração Parametrizável de Scripts de Teste

Garantia da Cobertura de Testes

Mix de Técnicas do Estado-da-Arte

Cobertura de Regras de Sistema (derivadas de Regras de Negócio)

Geração Combinatória com Otimizações de Caminhos e Dados

Gerações Aleatórias Baseadas em Semente Parametrizável

Algoritmos Parametrizáveis

Facilidade de Automação e Uso

Procura instalar dependências necessárias automaticamente

Gera configuração padrão para frameworks de teste

Tenta abstrair execução de ferramentas externas

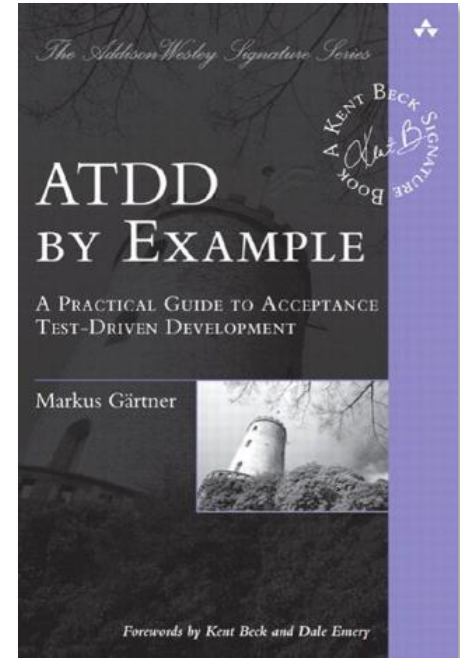
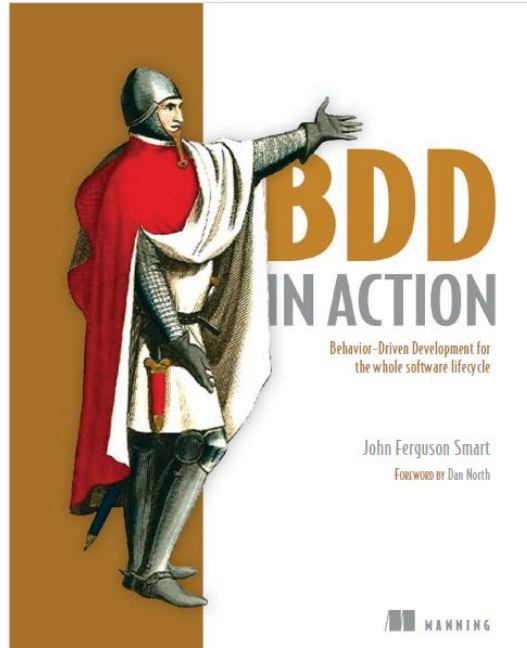
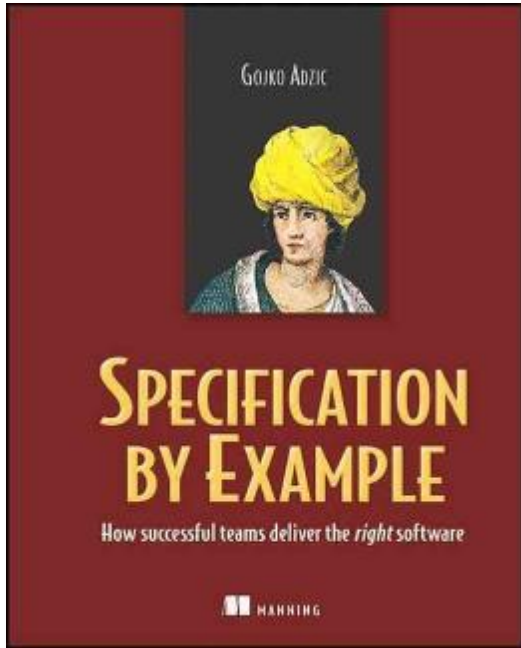
contextualização

“No desenvolvimento ágil, é importante que o desenvolvimento da documentação de usuário seja parte do mesmo processo que o ciclo de vida do produto, e seja executado em conjunção com o desenvolvimento do software. [...]”

“Isso permite que o **software** e a **documentação** de usuário sejam testados, distribuídos e mantidos **juntos**. [...]”

“No desenvolvimento ágil, o software não pode ser considerado completo sem a produção e a validação da documentação de usuário associada.”

SbE/BDD/ATDD



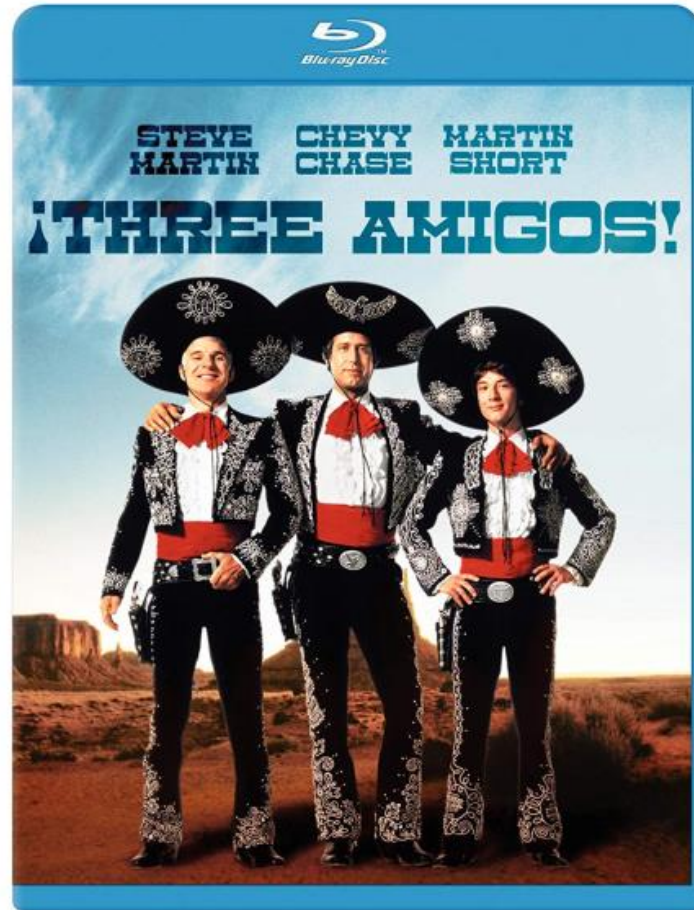
Documentação de Usuário é **artefato primário**

- discutir e validar requisitos com interessados
- criar um entendimento compartilhado
- derivar testes

desenvolvimento colaborativo

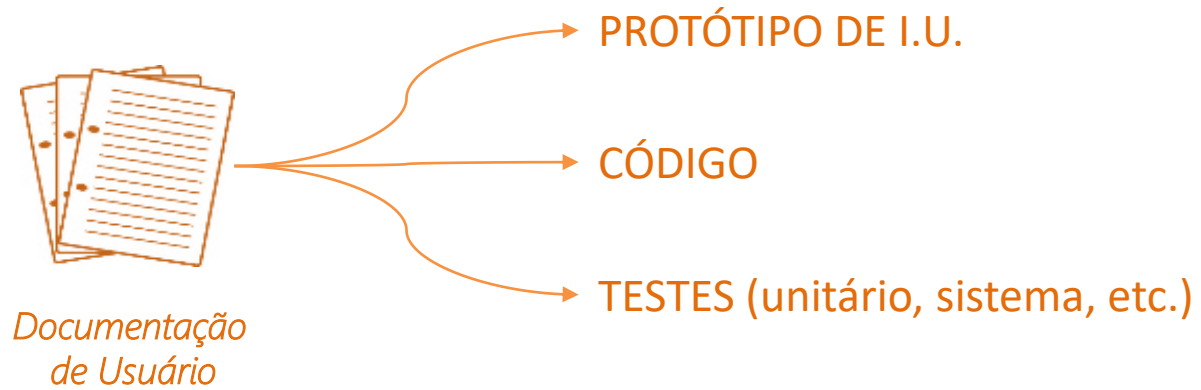


analista de negócio, analista de sistemas, projetista de iu, projetista de ux, desenvolvedor, testador



analista, programador, testador

exemplo



DOCUMENTAÇÃO DE USUÁRIO

Funcionalidade: Aumentar Salário

Como um funcionário do RH

Eu desejo poder aumentar percentualmente o salário de um funcionário

Para mantê-lo motivado

Cenário: Aumenta salário percentualmente

Dado que estou na tela de aumento de salário

Quando escolho o funcionário

e informo o percentual do aumento

e confirmo o aumento

Então eu vejo uma mensagem informando que o aumento foi concedido

Cenário: Não deve conceder aumento acima do percentual máximo

Dado que estou na tela de aumento de salário

Quando escolho o funcionário

e informo um percentual de aumento acima de 20%

e confirmo o aumento

Então eu vejo uma mensagem informando que o aumento está acima do limite permitido



*Documentação
de Usuário*

→ **PROTÓTIPO DE I.U.**

Funcionário:

nome

BUSCAR

Percentual:

0.00

AUMENTAR

→ **CÓDIGO**

→ **TESTES**



*Documentação
de Usuário*

TESTE UNITÁRIO

```
describe( "raiseSalary", () => {  
  ...  
  
  it( "should not exceed maximum", async () => {  
    const percentage = MAX_SALARY_PERCENTAGE + 0.01;  
    expect( async() => {  
      await service.raiseSalary( EMPLOYEE_ID, percentage );  
    } ).toThrow( BusinessError );  
  } );  
  
} );
```

CÓDIGO

```
async function raiseSalary( employeeId: number, percentage: number): Promise<boolean> {  
  if ( percentage > MAX_SALARY_PERCENTAGE ) {  
    throw new BusinessError( this.i18n.msg( SALARY_RAISE_MAX_EXCEEDED, percentage ) );  
  }  
  return await this.db.exec( UPDATE_SALARY_SQL, { id: employeeId, percentage:  
percentage } ) > 0;  
}
```



*Documentação
de Usuário*

TESTE FUNCIONAL (I.U.)

```
test( "salary should not exceed maximum", async ( t ) {  
  const percentage = MAX_SALARY_PERCENTAGE + 0.01;  
  const msg = i18n.msg( SALARY_RAISE_MAX_EXCEEDED, percentage );  
  await t  
    .typeText( "#employeeName", "José da Silva" )  
    .click( "#search" )  
    .typeText( "#percentage", percentage )  
    .click( "#raise" )  
    .expect( Selector( "#msg" ).innerText )  
    .eq( msg );  
} );
```

RECAPTULANDO

Documentação de usuário como artefato agregador

Mídia para discussão entre a equipe e equipe + clientes

Compreensão comum, Validação

Prototipação, Desenvolvimento, Testes, Feedback

como
escrever uma
doc. de usuário

O QUE SE USA?

Vários mapeamentos sistemáticos da literatura
(CURCIO et al., 2018; INAYAT et al., 2015;
SCHÖN; THOMASCHEWSKI & ESCALONA, 2017)
apontam que **empresas ágeis** usam **Funcionalidades**

 **Pesquisas (surveys) de mercado indicam o mesmo**

CURCIO, Karina; NAVARRO, Tiago; MALUCELLI, Andreia; REINEHR, Sheila. Requirements engineering: A systematic mapping study in agile software development. Journal of Systems and Software, 2018.

INAYAT, Irum; SALIM, Siti Salwah; MARCZAK, Sabrina; DANEVA, Maya; SHAMSHIRBAND, Shahaboddin. A systematic literature review on agile requirements engineering practices and challenges Computers in Human Behavior, 2015.

SCHÖN, Eva-Maria; THOMASCHEWSKI, Jörg; ESCALONA, María José. Agile Requirements Engineering: A systematic literature review. Computer Standards & Interfaces, 2017.

Funcionalidade: Aumentar Salário

Como um funcionário do RH

Eu desejo poder aumentar percentualmente o salário de um funcionário

Para mantê-lo motivado

Cenário: Aumenta salário percentualmente

Dado que estou na tela de aumento de salário

Quando escolho o funcionário

e informo um percentual do aumento inferior ou igual a 20%

e confirmo o aumento

Então o aumento é concedido

e eu vejo uma mensagem informando que o aumento foi concedido

Cenário: Não deve conceder aumento acima do percentual máximo

Dado que estou na tela de aumento de salário

Quando escolho o funcionário

e informo um percentual de aumento acima de 20%

e confirmo o aumento

Então eu vejo uma mensagem informando que o aumento está acima do limite permitido

FUNCIONALIDADE

Funcionalidade: Aumentar Salário

Como um funcionário do RH

Eu desejo poder aumentar percentualmente o salário de um funcionário

Para mantê-lo motivado

modelo de funcionalidade



Como um <PAPEL>

Eu desejo <FUNCIONALIDADE DESEJADA>

Para <OBJETIVO DE NEGÓCIO ou BENEFÍCIO>

CENÁRIO

Cenário: Aumenta salário percentualmente

Dado que estou na tela de aumento de salário

Quando escolho o funcionário

e informo um percentual do aumento inferior ou igual a 20%

e confirmo o aumento

Então o aumento é concedido

e eu vejo uma mensagem informando que o aumento foi concedido

modelo de cenário



Dado que <PRÉ-CONDIÇÃO>

Quando <AÇÃO>

Então <RESULTADO ou PÓS-CONDIÇÃO>

MODELO DE CENÁRIO

Escreva uma única <PRÉ-CONDIÇÃO>, <AÇÃO> ou <RESULTADO/PÓS-COND. > por linha

Use “e” na linha seguinte quando houver mais

```
Dado que <PRÉ-CONDIÇÃO 1>  
    e <PRÉ-CONDIÇÃO 2>  
Quando <AÇÃO 1>  
    e <AÇÃO 2>  
Então <RESULTADO/PÓS-COND. 1>  
    e <RESULTADO/PÓS-COND. 2>
```

definições

DEFINIÇÕES

“Uma **Funcionalidade** (*feature*) é um fragmento de comportamento do software que ajuda usuários ou outros interessados a atingir alguma meta de negócio.”

(SMART, 2014)

“Um **Cenário** é uma descrição passo-a-passo de uma série de eventos que ocorrem sequencialmente ou concorrentemente.”

(ISO; IEC & IEEE, 2017)

DEFINIÇÕES

DECLARADOS

“Um **Cenário Positivo** (ou Cenário de Sucesso) é aquele que **produz pós-condições**, devido a todas as entradas serem consideradas **corretas** (válidas).”

“Um **Cenário Negativo** (ou Cenário de Tratamento de Erro) é aquele que **não** produz pós-condições, devido a uma ou mais entradas – *i.e.*, ações ou dados – **inválidas** terem sido usadas.”

“Um **Cenário de Falha** é aquele que **não** produz pós-condições devido a **defeitos inesperados**.”

Funcionalidade: Aumentar Salário

Como um funcionário do RH

Eu desejo poder aumentar percentualmente o salário de um funcionário

Para mantê-lo motivado

Cenário: Aumenta salário percentualmente

Dado que estou na tela de aumento de salário

Quando escolho o funcionário

e informo um percentual do aumento inferior ou igual a 20%

e confirmo o aumento

Então o aumento é concedido ← PÓS-CONDIÇÃO

e eu vejo uma mensagem informando que o aumento foi concedido

Cenário Positivo

Cenário: Não deve conceder aumento acima do percentual máximo

Dado que estou na tela de aumento de salário

Quando escolho o funcionário

e informo um percentual de aumento acima de 20%

e confirmo o aumento

Então eu vejo uma mensagem informando que o aumento está acima do limite permitido

Cenário Negativo

DEFINIÇÕES

“Uma **Pré-condição** é um **estado** do sistema **requerido** para que um cenário possa ocorrer.”

“Uma **Pós-condição** é um **estado produzido** por um cenário do sistema.”

É comum uma Pré-condição ter sido produzida por outro cenário – ou seja, ser uma Pós-condição dele.

Funcionalidade: Aumentar Salário

Cenário: Aumenta salário percentualmente

Dado que eu tenho o **acesso ao sistema concedido** ← PRÉ-CONDIÇÃO
e estou na tela de aumento de salário

Quando escolho o funcionário
e informo um percentual do aumento inferior ou igual a 20%
e confirmo o aumento

Então o aumento é concedido
e eu vejo uma mensagem informando que o aumento foi concedido

Funcionalidade: Efetuar Login

Cenário: Login efetuado com sucesso

Dado que estou na tela de login
Quando informo minhas credenciais
e confirmo a entrada no sistema

Então eu tenho o **acesso ao sistema concedido** ← PÓS-CONDIÇÃO
e eu vejo uma mensagem de boas vindas

RECAPTULANDO

Funcionalidades são largamente usadas para documentar vários tipos de software

São simples de ler e validar

São fáceis de escrever e possuem modelos

Não requerem um editor de texto especial

Manutenção simples



perguntas