



BCC

Engenharia de Software

2018.2

Professor Rodrigo Andrade

Requisitos

O que os stakeholders
desejam da aplicação?

Requisitos

- Declaram funcionalidades, restrições, condições e desejos dos stakeholders
- Funcionais e não-funcionais

Requisitos funcionais

- Declaram serviços ou funcionalidades que a aplicação deve fornecer
- Como o sistema deve reagir a uma determinada entrada
- Como o sistema deve se comportar em determinada situação

Requisitos não-funcionais

- Declaram **restrições** de serviços ou funções fornecidas pela aplicação
- Definem propriedades do sistema
- Podem ser até mais críticos do que requisitos funcionais
 - Se não satisfeitos, podem tornar o sistema inútil

O que é funcional e o que não é funcional?

Usuário deve poder buscar determinado artigo

Múltiplos usuários devem poder visualizar informações de artigos

Informações de um artigo devem ser armazenadas no BD MySQL

O tempo para armazenar um novo artigo deve ser de no máximo 30 segundos

Um relatório contendo os artigos que foram criados deve ser disponibilizado aos usuários quando requisitado

Requisitos “tradicionais”

ECLIPSE/Workstation/Tools/DE/FS/3.5.1

Function Add node

Description Adds a node to an existing design. The user selects the type of node, and its position. When added to the design, the node becomes the current selection. The user chooses the node position by moving the cursor to the area where the node is added.

Inputs Node type, Node position, Design identifier.

Source Node type and Node position are input by the user, Design identifier from the database.

Outputs Design identifier.

Destination The design database. The design is committed to the database on completion of the operation.

Requires Design graph rooted at input design identifier.

Pre-condition The design is open and displayed on the user's screen.

Post-condition The design is unchanged apart from the addition of a node of the specified type at the given position.

Side-effects None

Definition: ECLIPSE/Workstation/Tools/DE/RD/3.5.1

Casos de uso

- Descrevem o comportamento do sistema sob várias condições
- Conta uma história sobre como um usuário interage com o sistema
- Descreve o sistema do ponto de vista do usuário

[UC_010] Alterar senha

Descrição: Este caso de uso tem como finalidade alterar a senha do usuário no sistema.

Pré-condições: O usuário estar logado no sistema.

Pós-condições: Não há.

Fluxo de eventos principal

1. Este caso de uso inicia quando o usuário seleciona a opção de alterar a senha.
2. O usuário preenche os campos de senha atual e nova senha.
3. O sistema valida as informações fornecidas [FS20] [FS21]
4. O usuário é tem sua senha alterada.

Campos da Lista

Nome	Descrição	Tipo	Tam.	Restrições
Senha atual	A senha que o usuário deseja alterar	Lista	8	Campo Obrigatório;
Nova senha	Nova senha do usuário	Lista	8	Campo Obrigatório;

Fluxos secundários (alternativos e de exceção)

[FS20] Campos obrigatórios não preenchidos

10. O sistema verifica se os campos obrigatórios foram preenchidos, caso não tenham sido preenchidos o sistema emite uma mensagem “Um ou mais campos obrigatórios não foram preenchidos!”.

[FS21] Senha inválida

11. O sistema verifica se a senha informada pertence ao usuário logado no sistema. Caso não pertença, o sistema emite uma mensagem: “Senha inválida!”

[UC_008] Logar no sistema

Descrição: Este caso de uso tem como finalidade logar um usuário no sistema.

Pré-condições: Não há.

Pós-condições: Não há.

Fluxo de eventos principal

1. Este caso de uso inicia quando o usuário inicializa o sistema.
2. O usuário preenche os campos de login e senha e seleciona a opção de logar.
3. O sistema valida as informações fornecidas [FS18] [FS19]
4. O usuário é logado no sistema.

Campos da Lista

Nome	Descrição	Tipo	Tam.	Restrições
Login	Login do usuário	Lista	10	Campo Obrigatório;
Senha	Senha do usuário	Lista	8	Campo Obrigatório;

Fluxos secundários (alternativos e de exceção)

[FS18] Campos obrigatórios não preenchidos

8. O sistema verifica se os campos obrigatórios foram preenchidos, caso não tenham sido preenchidos o sistema emite uma mensagem “Um ou mais campos obrigatórios não foram preenchidos!”.

[FS19] Usuário ou Senha inválida

9. O sistema verifica se a senha informada pertence ao usuário informado. Caso não pertença, o sistema emite uma mensagem: “Usuário e senha não conferem!”

Abordagem para ciclo de vida ágil

Relembrando ciclo de vida ágil

- Trabalha de maneira próxima e contínua aos stakeholders
- Desenvolvimento dividido em iterações
- Projeto guiado por comportamento (BDD)

BDD

- Questiona o comportamento de uma aplicação antes e durante o desenvolvimento
 - Visa evitar falhas de comunicação
- Requisitos continuamente aprimorados
- Requisitos são **Histórias de usuários**

Histórias de usuários

- Descrevem como se espera que a aplicação seja usada
- Versão leve de requisitos
- Ajudam stakeholders a planejar e priorizar o desenvolvimento
- Ponto de partida para o desenvolvimento
- Foco no comportamento da aplicação ao invés da implementação

Feature: Artigo

As an usuario do sistema Blog

I want to adicionar, remover, ver e editar um artigo
so that eu nao ter que fazer isso manualmente

Formato das histórias de usuários

Feature [nomeFeature]

As a [algun stakeholder]

I want to [executar alguma tarefa]

so that [alcançar um objetivo]

Histórias de usuários SMART

- História boa x história ruim
- Específico
- Mensurável
- Realizável
- Relevante
- Duração fixa

Específico

Feature: Usuário pode buscar um artigo

Feature: Usuário pode buscar um artigo por título

Mensurável

Feature: Tempo de resposta para uma busca a artigo por titulo deve ser mínimo

Feature: Tempo de resposta para uma busca a artigo por titulo deve ser de no máximo 30 segundos

Realizável

- História de usuário completamente implementada em uma iteração
- Se não der tempo, é um indício de que as histórias devem ser quebradas

Relevante

- Perguntar "Por quê?" constantemente
 - Por que adicionar um artigo?
 - Por que buscar um artigo por título?
 - Por que gerar relatório de postagens para usuários?
 - ...
- Responder sempre as perguntas

Duração fixa

- Interrompe o desenvolvimento da história caso o tempo tenha excedido a iteração
- Quebra-se em histórias menores e implementa na próxima iteração
- Aprender a estimar o tempo de realização

Qual ferramenta podemos
testar histórias de
usuários?

Cucumber

Ferramenta Cucumber

- Transforma Histórias de usuários em
 - Testes de aceitação
 - Testes de integração
- Uma funcionalidade, vários **cenários**
 - Cenários são compostos por **passos**



Exemplo de cenário do Cucumber

Feature: Novo artigo

Scenario: novo artigo

Given Eu abro a pagina inicial de artigo

When Eu crio o artigo "Ruby eh uma excelente linguagem"
com titulo "Ruby"

Then Eu vejo que "Ruby" foi salvo

Passos

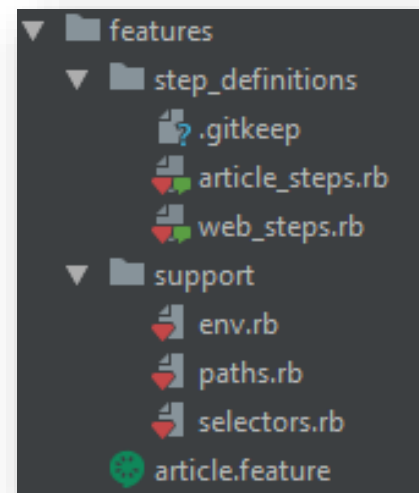
- **Given**
 - Geralmente estabelece pré-condições
- **When**
 - Simula alguma ação do usuário, como pressionar um botão
- **Then**
 - Verifica se alguma condição foi satisfeita
- **And e But**
 - Permite versões mais complexas dos passos acima

Definições do passo

- Código Ruby em um arquivo separado

```
Then(~/^Eu vejo que "([^"]*)" foi salvo$/ ) do |name|  
  // codigo Ruby do teste  
end
```

Regexes: Combinam as frases escritas nos passos com a definição do conjunto de testes



Cenários baseados em GUI

Definições

- Testam o uso do sistema de acordo com o usuário
- Focam nas funcionalidades que o usuário pode ver
- Tenta simular o **usuário usando o sistema**
- Não focam em mudanças de estado internas

Exemplo: criando artigo inválido

Scenario: novo artigo invalido

Given Eu estou na pagina de artigo

When Eu preencho o texto "" e o titulo "invalid"

And Eu clico em criar artigo

Then Eu vejo uma mensagem de erro do novo artigo

Exemplo: removendo um artigo

Scenario: remover um artigo existente

Given Eu estou na pagina de artigo

And Eu vejo o titulo "titulo" do artigo

When Eu clico para ver "titulo"

And Eu clico remover o artigo "titulo"

Then Eu vejo uma mensagem que "titulo" foi removido

Qual o problema do cenário?

Scenario: novo artigo

Given o sistema nao tem um artigo com titulo "titulo"

When Eu crio um artigo com titulo "titulo"

Then o artigo com titulo "titulo" é armazenado corretamente no sistema

Como simular um usuário?

Capybara

- Fornece uma DSL para descrever ações do usuário
- Simula um usuário usando a GUI
- Permite implementar testes de GUI

Capybara



Exemplo

```
Given('the article entitled {string} with text {string}
exists') do |title, body|
  visit 'articles/new'
  expect(page).to have_content('New article')
  fill_in 'article[title]', :with => title
  fill_in 'article[text]', :with => body
  click_button 'submit'
  expect(page).to have_content(title)
end
```

Resumo da aula

- Requisitos
 - Funcionais e não-funcionais
- Histórias do usuário
- Desenvolvimento baseado em comportamento



Referência



Capítulo 7