

Universidade Federal do Espírito Santo
Centro Tecnológico
Departamento de Informática



Disciplina: INF15978 – Engenharia de Software I

Prof.: **Monalessa Perini Barcellos**

(monalessa@inf.ufes.br)

1

User Stories

Como um
<PAPEL>
eu posso/gostaria/devo —————> termo mais geral:
<FUNÇÃO> **quero**
para/de
<RESULTADO para o NEGÓCIO>

Como um vendedor, gostaria de consultar o estoque de um determinado produto para oferecer a um cliente

Como um vendedor, devo registrar a venda realizada para um cliente para manter o histórico de vendas

Como um diretor, gostaria de obter o volume de vendas do mês para acompanhar o atingimento das metas

2

User Stories

Para definir boas user stories

INVEST

I Independent
N Negotiable
V Valuable to users or customers
E Estimatable
S Small
T Testable

User Stories

Writing a user story

- 1 Define your **end user**
- 2 Specify what **they want**
- 3 Describe **the benefit**
- 4 Add **acceptance criteria**

User Stories

Critérios de Aceitação

Complementam a user story e indicam os principais critérios que devem ser considerados para testá-la. São importantes para considerar a user story “pronta” (“done”).

Como um vendedor, gostaria de consultar o estoque de um determinado produto para oferecer a um cliente

CA1: Deverão ser exibidos para um dado código de produto: estoque atual, estoque mínimo, nome do local de estoque

CA2: Nome do local do estoque com estoque abaixo do mínimo deve aparecer em vermelho

CA3: O tempo de resposta não pode ultrapassar 2 segundos

User Stories

Outros exemplos

User story: As a user, I want to be able to register online, so that I can start shopping online.

Acceptance criteria:

- ☐ User can only submit a form by filling in all required fields
- ☐ The email user provided must not be a free email
- ☐ Submission from same IP can only be made three times within 30 minutes
- ☐ User can only submit a form by filling in all required fields
- ☐ User will receive a notification email after successfully registration

As an Account Manager
I want a sales report of my account to be sent to my inbox daily
So that I can monitor the sales progress of my customer portfolio

Acceptance criteria:

1. The report is sent daily to my inbox
2. The report contains the following sales details: ...
3. The report is in csv format.

Customer Inter Account Transfer

As a bank customer,
I want to transfer funds between my linked accounts,
So that I can fund my credit card.

Acceptance Criterion 1:

Given that the account is has sufficient funds
When the customer requests an inter account transfer
Then ensure the source account is debited
AND the target account is credited.

Acceptance Criterion 2:

Given that the account is overdrawn,
When the customer requests an inter account transfer
Then ensure the rejection message is displayed
And ensure the money is not transferred.

User Stories

Para facilitar e padronizar a documentação, pode-se definir um template específico para cada situação.

Exemplo:

Para cadastros simples

USX. Como <<ator>> quero cadastrar <<objeto>> para <<finalidade>>	CAX.1 Para cadastrar um <<objeto>> devem ser informados <<campos>>
	CAX.2 Um <<objeto>> não pode ter as seguintes informações alteradas <<campos>>
	CAX.3 A consulta pode ser feita por <<campos>>
	CAX.4 <<Objetos>> relacionados a <<outros objetos>> não podem ser excluídos
	CAX.5 Alterações ou exclusões devem ser comunicadas e deve ser requerida confirmação para efetivar

Para consultas

USX. Como <<ator>> quero consultar <<informações>> para <<finalidade>>	CAX.1 Para realizar a consulta, o <<ator>> deve informar como parâmetros <<parâmetros>>
	CAX.2 Devem ser apresentadas as seguintes informações para o usuário <<campos>>

Engenharia de Software

Monalessa Perini Barcellos

7

User Stories

User Stories são apenas o primeiro passo...



Engenharia de Software

Monalessa Perini Barcellos

8

User Stories x Casos de Uso

- Apesar de uma user story deixar claro que uma funcionalidade é requisitada no sistema por seus usuários, ela não dá ideia de como a funcionalidade vai ser (i.e., como o sistema vai se comportar), enquanto que os casos de uso dão essa ideia. No geral, uma user story é só uma forma de registrar a existência de um requisito, sem muitos detalhes, enquanto que um caso de uso é a documentação do resultado de se pensar em como aquele requisito vai ser tratado pelo sistema na prática.
- User stories são centradas no resultado e no benefício da coisa que se está descrevendo, enquanto casos de uso são mais granulares e descrevem como o sistema irá agir.
- User stories são apenas o início do processo de entendimento dos requisitos do sistema. Para os desenvolvedores que implementarão o sistema, casos de uso parecem um jeito muito melhor de descrever requisitos do que user stories. Mas, descrever casos de uso demanda esforço.
- Boa prática: não descrever casos de uso até que a necessidade pela informação que eles apresentam seja concretamente percebida.

Engenharia de Software

Monalessa Perini Barcellos

9

Combinando User Stories e Casos de Uso

- As user stories representam os requisitos do usuário e critérios de aceitação. Então, os requisitos devem ser inicialmente representados como user stories.
- Como não há diagrama para representar as user stories, o diagrama de casos de uso pode ser usado para fornecer uma visão comportamental do sistema. Para isso, cada user story pode ser representada como um caso de uso. Porém, dependendo do nível de granularidade da user story, é possível que mais de uma user story seja englobada em um caso de uso.
- Para user stories mais complexas ou que precisam de descrição mais detalhada para serem implementadas corretamente, pode-se fazer a descrição do caso de uso.

Nota: em projetos ágeis é menos comum fazer esse detalhamento formalmente. Geralmente, se usa wireframes e mockups (muitas vezes, com anotações) para apresentar as funcionalidades e capturar as informações sobre como o sistema deve se comportar para implementar a referida user story. Embora não seja a mesma coisa que descrever um caso de uso, pode servir ao propósito e não levar à necessidade de uma descrição detalhada do caso de uso.

Engenharia de Software

Monalessa Perini Barcellos

10

Referências

- SCHWABER, K., 2004, Agile Project Management with Scrum, Paperback.
- <https://pt.slideshare.net/manoelp/exemplos-de-user-stories>
- <https://dev.to/code2bits/user-story-guidelines-l2e>
- <https://www.visual-paradigm.com/scrum/definition-of-done-vs-acceptance-criteria/>
- <https://agileusa.wordpress.com/2018/03/03/how-to-write-a-user-story/>
- <https://www.knowledgetrain.co.uk/agile/how-to-write-a-user-story>
- <https://pt.stackoverflow.com/questions/134291/mapeamento-entre-user-stories-e-casos-de-uso>
- <https://blog.casecomplete.com/post/Agile-Use-Cases-in-Four-Steps>

Engenharia de Software

Monalessa Perini Barcellos

11

Universidade Federal do Espírito Santo
Centro Tecnológico
Departamento de Informática



Disciplina: INF15978 – Engenharia de Software I

Prof.: Monalessa Perini Barcellos

(monalessa@inf.ufes.br)

12

O sistema deve permitir o registro de agendamento de serviços, que são registrados com status “agendado”. Para realizar um agendamento, a recepcionista deve informar o nome do cliente, o animal, seu tipo (cão ou gato), seu tamanho (pequeno, médio ou grande), o serviço, o horário para realização do serviço e o funcionário alocado. Quando o serviço é iniciado, o horário de início é registrado e o agendamento deve mudar para o status “iniciado”. Quando o serviço é finalizado, o horário de término é registrado e o status do agendamento muda para “concluído”. Caso o atraso para início do agendamento seja superior a 30 minutos, o agendamento deve aparecer em vermelho na tela e o sistema deve enviar uma mensagem para o celular do cliente informando. O envio da mensagem deve durar no máximo 5 segundos.

13

Descrevendo requisitos na forma “tradicional” (como feito nas atividades anteriores)

Requisitos Funcionais:

RF01: O sistema deve permitir o registro de agendamento de serviços, devendo ser informados o nome do cliente, o animal, seu tipo (cão ou gato), seu tamanho (pequeno, médio ou grande), o serviço, o horário para realização do serviço, o funcionário alocado, o horário de início e o horário de fim. (RN01, RN02, RN03)

RF02: O sistema deve ser capaz de enviar mensagem para os clientes. (RN04)

Regras de Negócio:

RN01: Quando um agendamento é criado, ele é registrado com o status “agendado”.

RN02: Quando o serviço é iniciado, o horário de início é registrado e o agendamento recebe o status “iniciado”.

RN03: Quando o serviço é finalizado, o horário de término é registrado e o agendamento recebe o status “concluído”.

RN04: O sistema deve enviar uma mensagem informando para o cliente caso haja um atraso superior a 30 minutos em seu agendamento.

Requisitos Não Funcionais:

RNF01: Agendamentos com atraso superior a 30 minutos devem aparecer em vermelho na tela (categoria: usabilidade, escopo: funcionalidade RF01).

RNF02: O envio de mensagens para o celular do cliente deve levar até 5 segundos (categoria: eficiência, escopo: funcionalidade RF02)

14

Descrevendo requisitos na forma de user story

User Stories

Eu como recepcionista, quero controlar o agendamento de serviços para atender os clientes.

Critérios de aceitação:

CA1. Para registrar um agendamento, devem ser informados o nome do cliente, o animal, seu tipo (cão ou gato), seu tamanho (pequeno, médio ou grande), o serviço, horário para realização do serviço, o funcionário alocado.

CA2. Quando um agendamento é criado, ele é registrado com o status “agendado”.

CA3. Quando o serviço é iniciado, o horário de início deve ser registrado e o agendamento recebe o status “iniciado”.

CA4. Quando o serviço é finalizado, o horário de término deve ser registrado e o agendamento recebe o status “concluído”.

CA5. Quando há atraso superior a 30 minutos no início do serviço agendado, o agendamento deve aparecer em vermelho na tela.

CA6. Quando há atraso superior a 30 minutos no agendamento, o sistema envia mensagem para o cliente.

CA7. O tempo de envio da mensagem para o celular do cliente deve ser de até 5 segundos.