

Histórias de Usuários

- Conceitos básicos e exemplos
- Aplicação ao projeto final

Referências

- COHN, Mike. Desenvolvimento de Software com SCRUM. Porto Alegre: Bookman, 2011.
- VALENTE, Marco Túlio O. Engenharia de Software Moderna. Belo Horizonte: 2020.

O que é uma História de Usuário?

- “Uma história de usuário é uma descrição curta e simples de um requisito contada a partir da perspectiva da pessoa que deseja a nova funcionalidade do sistema”. (COHN, p.258-259, 2011)
- Devem ser escritas de maneira a serem compreendidas por usuários e desenvolvedores
- Detalhes do comportamento do sistema não aparecem na história
- Veja um modelo simples:

Como <tipo de usuário>, desejo <algum objetivo> para que <alguma razão>.

História e Requisito de Usuário

- Ambos são curtas, fáceis de ler e compreensíveis a desenvolvedores, partes interessadas e usuários
- A História de Usuário tem foco no valor definido pelo usuário
- O Requisito de Usuário apresenta um serviço ou restrição do software com foco no negócio
- A História de Usuário facilita o entendimento do valor para cada *persona*

Um outro modelo de histórias de usuários

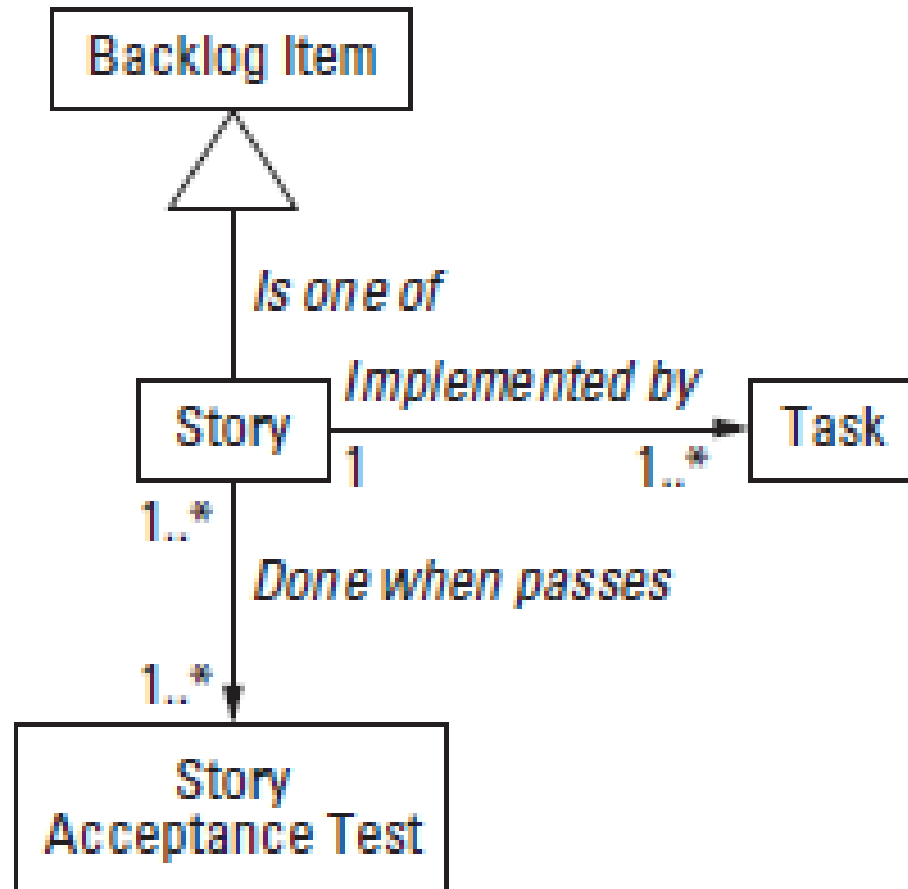
***Como um* <papel>, *eu posso/devo* <atividade>
para que <valor para o negócio>**

- **Papel**
 - Permite uma segmentação da funcionalidade do produto
 - Identifica necessidades baseadas em papéis
- **Atividade**
 - Representa “requisito do sistema”
- **Valor para o negócio**
 - Comunica porque a atividade é necessária
 - Atividades alternativas podem ser discutidas

Quem deve escrever?

- No XP, a proposta era o próprio usuário escrever suas histórias
- No Scrum, podem ser escritas pelo PO ou em conjunto com os usuários
- Na prática, geralmente, qualquer um pode escrever uma história, contudo, o PO do Scrum *escreve, revisa e prioriza* o conjunto de histórias de usuários

Uma visão de banco de dados



Representação das Histórias

- Cartão
 - Representa duas ou três frases usadas para descrever a intenção da história
 - Detalhe ainda será determinado
 - Promessa de conversa
- Conversa
 - Interação contínua entre a equipe, cliente e PO para determinar comportamento detalhado do software
- Confirmação
 - Representa o teste de aceitação
 - Representa condições de satisfação

Detalhamento das Histórias

- Detalhes são alcançados principalmente pela conversação
- Se mais detalhes são necessários, podem ser providos por anexos
 - Mock-up, planilha ou algoritmos
- Detalhes são coletados na medida em que se tornam necessários (*just in time*)

Um exemplo: Loja Virtual (VALENTE, 2020)

- Cartão: "Fechar uma compra"
- Conversas: PO explica os meios de pagamento; as formas de entrega, formas de parcelamento, etc.
- Confirmação:
 - Testar compra à vista e compra parcelado
 - Testar com cartões de crédito A, B e C
 - Testar com modos de entrega X e Y

Um exemplo: Loja Virtual (VALENTE, 2020)

- Cartão: "Fechar uma compra"
- Conversas: PO explica os meios de pagamento; as formas de entrega, formas de parcelamento, etc.
- Confirmação:
 - Testar compra à vista e compra parcelado
 - Testar com cartões de crédito A, B e C
 - Testar com modos de entrega X e Y

Outro exemplo: Sistema de Controle de Biblioteca (VALENTE, 2020)

- 3 tipos de usuários
 - Usuário típico
 - Professor
 - Funcionário da biblioteca

Histórias do usuário típico (VALENTE, 2020)

- Como usuário típico, eu gostaria de realizar empréstimos de livros
- Como usuário típico, eu gostaria de devolver um livro que tomei emprestado
- Como usuário típico, eu gostaria de renovar empréstimos de livros
- Como usuário típico, eu gostaria de pesquisar por livros
- Como usuário típico, eu gostaria de reservar livros que estão emprestados
- Como usuário típico, eu gostaria de receber e-mails com novas aquisições

Histórias de professores (VALENTE, 2020)

- Como professor, eu gostaria de realizar empréstimos de maior duração
- Como professor, eu gostaria de sugerir a compra de livros
- Como professor, eu gostaria de doar livros para a biblioteca
- Como professor, eu gostaria de devolver livros em outras bibliotecas

Histórias de funcionários da biblioteca (VALENTE, 2020)

- Como funcionário, eu gostaria de cadastrar novos usuários
- Como funcionário, eu gostaria de cadastrar novos livros
- Como funcionário, eu gostaria de dar baixa em livros estragados
- Como funcionário, eu gostaria de obter estatísticas sobre o acervo
- Como funcionário, eu gostaria que o sistema envie e-mails de cobrança para alunos com empréstimos atrasados
- Como funcionário, eu gostaria que o sistema aplicasse multas quando da devolução de empréstimos atrasados

Características de boas histórias (INVEST)

- **I**ndependentes
- Abertas para **N**egociação
- Agregar **V**alor
- **E**stimáveis
- **S**ucintas
- **T**estáveis

Características de boas histórias

- Independentes
 - História pode ser desenvolvida, testada e até entregue isoladamente.
 - A conclusão de uma história deixa o produto em um estado em que ele pode ser entregue

Características de boas histórias

- **N**egociáveis
 - Não são contrato de implementação de algo
 - Oportunidade de discussão dos requisitos
 - Processo de colaboração
 - Deve haver possibilidade de *trade-offs* entre funcionalidade e datas de entrega
 - Deve haver flexibilidade para atender objetivos

Características de boas histórias

- Valor
 - Essência dos métodos ágeis
 - *Backlogs* são priorizados por valor
 - Mudança de uma abordagem de decomposição hierárquica funcional para abordagem vertical

Características de boas histórias

- Estimáveis

- Equipe deve ser capaz de prover uma estimativa de sua complexidade e quantidade de trabalho necessário para terminar a história
- Ao menos deve ser completada em uma iteração
- Se a equipe não consegue estimar, história deve ser quebrada em histórias menores
- Pode haver uso de “spikes” funcionais ou técnicos para aumentar assertividade e tomar decisão
- Conversa durante o processo de estimativa é importante para levantar requisitos implícitos
- Mudança de uma abordagem de decomposição hierárquica funcional para abordagem vertical

Características de boas histórias

- São de tamanho pequeno (*Small*)
 - Completadas em uma iteração
 - Histórias pequenas provêm maior agilidade
- Testáveis
 - Abordagem de se escrever o teste antes
 - TDD (*Test Driven Development*)
 - Se equipe sabe como testar uma história, eles sabem como codificá-la.
 - Palavras como “rapidamente”, “gerenciar”, “limpo” são fáceis de escrever mas ambíguas, devem ser evitadas