

Engenharia de Requisitos de Software

3º período

Professora: Michelle Hanne Soares de Andrade

mhsandrade@sga.pucminas.br

michellehanne.andrade@gmail.com

Sumário

- Manifesto Ágil
- SCRUM
- User Story

Manifesto Ágil

- Criado em fevereiro de 2001, por 17 profissionais, que já praticavam métodos ágeis como XP, DSDM, [Scrum](#), FDD.
- Os valores Manifesto Ágil são:

Indivíduos e interação entre
eles mais que processos e
ferramentas;

Software em funcionamento
mais que documentação
abrangente;

Colaboração do cliente mais
que negociação de
contratos;

Responder a mudanças mais
que seguir um plano.

Os doze princípios do Manifesto Ágil

- 1 - Nossa maior prioridade é **satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.**
- 2 - **Aceitar mudanças de requisitos**, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
- 3 - **Entregar frequentemente software funcionando**, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
- 4 - Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
- 5 - Construir projetos em torno de indivíduos motivados, dando a eles o ambiente e o suporte necessário e confiando neles para fazer o trabalho.

Os doze princípios do Manifesto Ágil

6 - O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é **por meio de conversa face a face**.

7 - **Software funcionando é a medida primária de progresso**.

8 - **Os processos ágeis promovem desenvolvimento sustentável**. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.

9 - **Contínua atenção a excelência técnica e bom design aumenta a agilidade**.

10 - **Simplicidade**: a arte de maximizar a quantidade de trabalho não realizado é essencial.

11 - **As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis**.

12 - Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

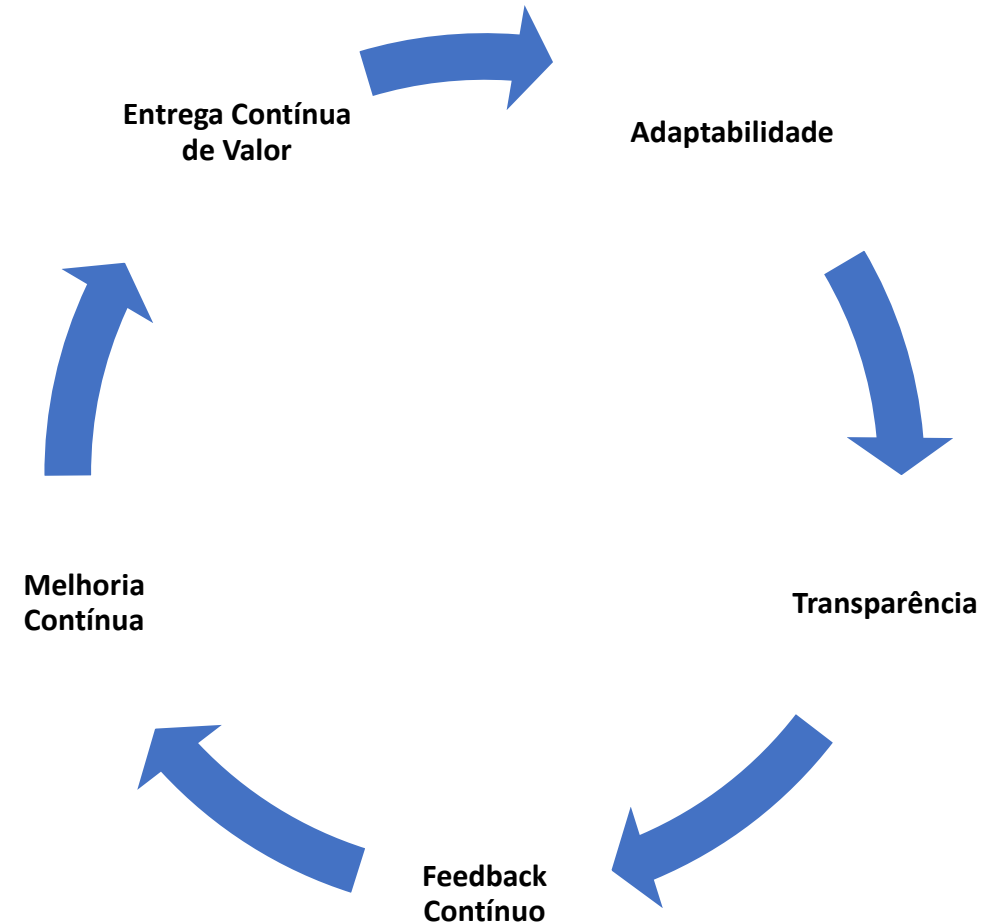
SCRUM

“Um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível.”

(Schwaber & Sutherland, 2017)

SCRUM

- “Scrum é sobre entregar valor aos clientes, sem necessariamente se importar no detalhe de como isso vai ser realizado” (Jeff Sutherland no Manifesto Ágil).



Fonte: o autor.

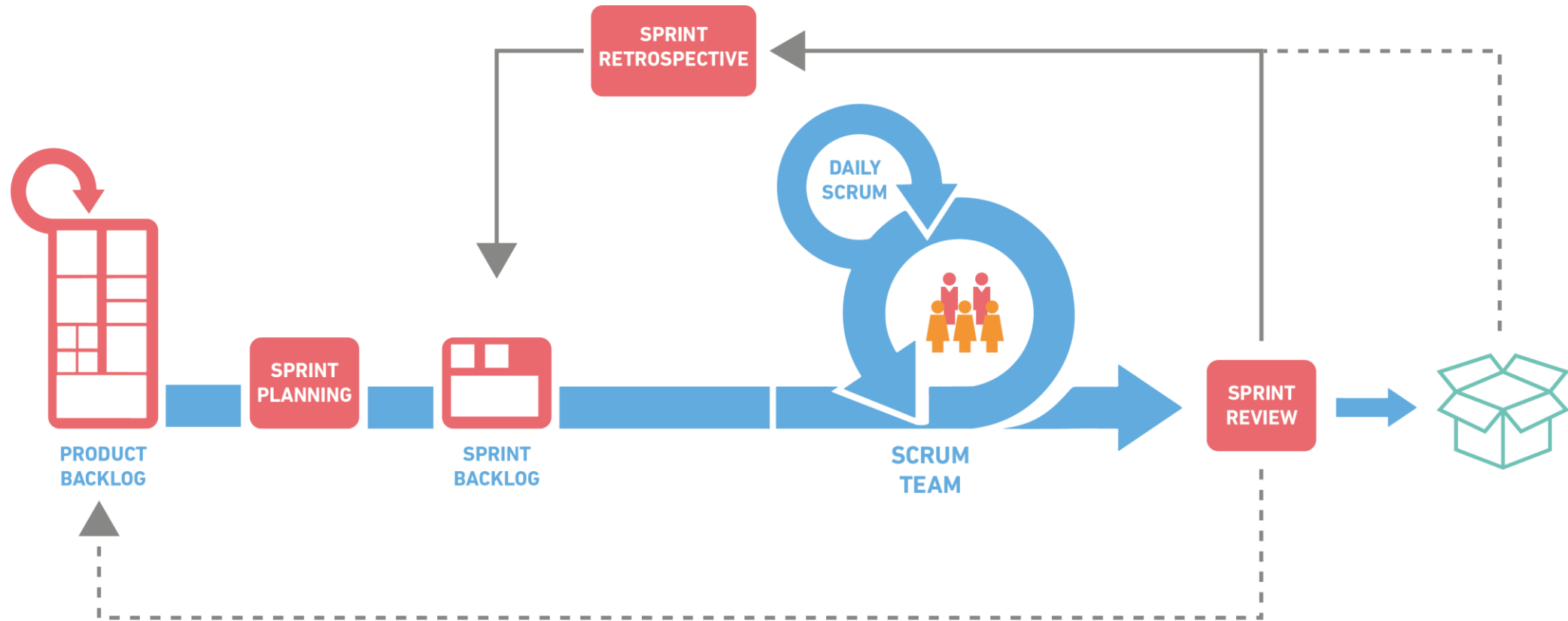
Só quem gosta de software é programador.
Cliente gosta de **solução que entrega valor**.
Se o seu time não entrega valor, ele não irá
muito longe. E se ele demora demais para
entregar valor, também não.

SCRUM

Utiliza ciclos de desenvolvimento - denominadas *sprints* - que permitem qualidade nas entregas e possibilidade de **mudança de requisitos ao longo do processo.**

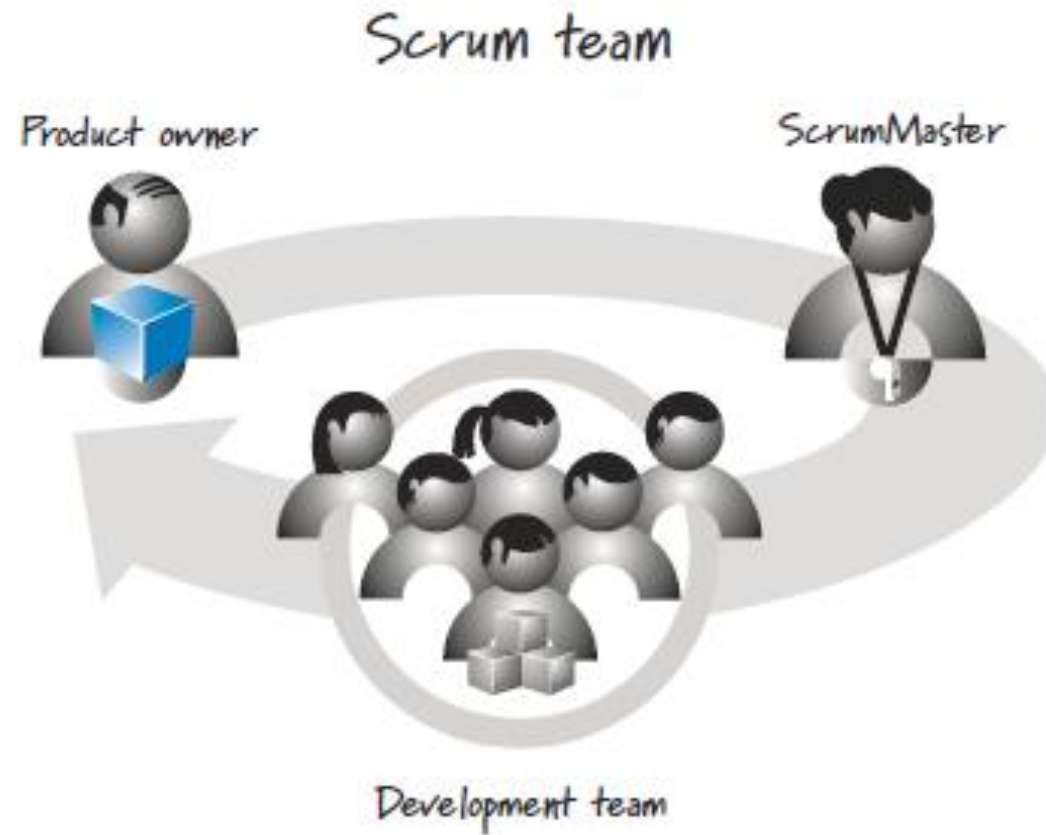
O framework sustenta-se em pilares e papéis bem definidos.

SCRUM



Fonte: <https://blog.mjv.com.br/frameworks-%C3%A1geis-saiba-como-funcionam-na-pratica>

SCRUM



Fonte: RUBIN, 2013.

SCRUM

Product Owner: É o responsável pelo ROI (retorno sobre o investimento) do projeto, por gerenciar e priorizar o Product Backlog e, para que o Scrum funcione realmente e ele consiga trabalhar, toda a organização deve respeitar as suas decisões.

Em alguns casos. o P.O. seria o Gerente de Produto (Product Manager), aquele que tem a visão, que define o roadmap, que colhe os insights de mercado e os transforma em requisitos para os desenvolvedores trabalharem.

SCRUM

DEVs: São os responsáveis por desenvolver o produto. O time de desenvolvedores (que não precisa e não deve ser composto apenas por desenvolvedores) é autogerenciável, multifuncional e compartilham a responsabilidade pelo sucesso ou fracasso do projeto.

O time de desenvolvimento é quem decide o quanto consegue entregar a cada iteração. Decide como será feita cada *feature*.

SCRUM

Scrum Master: É o responsável por aplicar e garantir a adoção do Scrum dentro da equipe. Cabe ao Scrum Master, que é um **líder-servidor**, liderar o time para que os objetivos do **Product Owner** sejam alcançados e para que o time de desenvolvimento consiga avançar sem impedimentos, removendo-os quando necessário.

Todos os eventos Scrum são facilitados pelo Scrum Master da equipe, que idealmente não deve ter outra função paralela no time Scrum.

SCRUM

- **Transparência** - todas as informações e atividades devem estar disponíveis para todos;
- **Inspeção** - as atividades são monitoradas, priorizadas e avaliadas periodicamente - em reuniões diárias (*daily*);
- **Adaptação** - após a inspeção, as atividades são refinadas e ajustadas para que não existam problemas. São realizadas reuniões de feedback para avaliar os pontos de mudança e o caminho a ser seguido em novas *sprints* (*plannings*).

SCRUM

Eventos: O Scrum chama seus eventos de time-boxes.

As sprints são time-boxes de 1 mês ou menos e são o coração do Scrum. Durante o período do Sprint um incremento utilizável do produto é criado.

Fazem parte do Sprint: planejamento, as reuniões diárias, a revisão e a retrospectiva.

Fonte: Duarte, Luiz. Scrum e Métodos Ágeis: Um Guia Prático . LuizTools. Edição do Kindle.

SCRUM

Os clientes se tornam parte da equipe de desenvolvimento e podem validar ou redefinir entregas.

Os riscos são melhores trabalhados e reduzidos, já que os progressos e atrasos são monitorados.

The Scrum Guide

Scrum Team:

- Product Owner
- Development Team
- Scrum Master

Artefatos:

- Product Backlog
- Sprint Backlog
- Increment

Eventos:

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Sprint Planning

Time-box de 8h para um sprint de um mês, ou menos tempo de acordo com o tamanho do Sprint. Nesta reunião é onde o Product Owner é ouvido em relação às prioridades e os objetivos deste Sprint. É nela também onde o time irá deliberar sobre o que conseguem fazer neste sprint em relação às necessidades do P.O., formalizando o **Sprint Backlog**, ou lista de coisas que serão feitas no próximo mês.

Product Backlog

- **Um Product Backlog nunca está completo.**
- O primeiro desenvolvimento do mesmo estabelece as características inicialmente conhecidas e requisitos melhor compreendidos.
- O **Product Backlog** evolui de acordo com o produto e o contexto em que será usado.
- O **Backlog do produto** é dinâmico; Constantemente sofre alterações para adequar o produto, ser competitivo e útil.

Product Backlog

O **Product Backlog** é uma lista ordenada de tudo o que se sabe ser necessário para o desenvolvimento do produto.

É a única fonte de requisitos para quaisquer alterações a serem feitas no produto.

O PO é responsável pelo **Product Backlog**, incluindo seu conteúdo, disponibilidade e controle de alterações.

Product Backlog

- Os itens do Backlog do produto geralmente incluem descrições de teste que comprovarão a integridade quando "Concluído".
- A equipe colabora nos detalhes dos itens do Backlog do Produto.
- Durante o refinamento do **Product Backlog**, itens são revisados. A equipe Scrum decide como e quando o refinamento é feito. **O refinamento geralmente consome não mais que 10% da capacidade da Equipe de Desenvolvimento.**
- No entanto, os itens do Backlog do produto podem ser atualizados a qualquer momento pelo PO.

Sprint Backlog

O Sprint Backlog é o conjunto de itens do Backlog do produto selecionados para o Sprint, visa o Incremento do produto e a meta do Sprint.

O Sprint Backlog é uma previsão dada pela equipe de desenvolvimento sobre qual será a funcionalidade da próxima entrega.

É necessário na entrega que as funcionalidade estejam com o status de "Concluído".

Daily Scrum

- **A Daily Scrum é um evento de 15 minutos** para o Time de desenvolvimento.
- A Daily Scrum é realizada todos os dias do Sprint.
- A equipe de desenvolvimento planeja o trabalho do dia.
- A Daily Scrum é usada para inspecionar o progresso em direção à Meta Sprint.

Daily Scrum

- **Aqui está um exemplo do que pode ser usado:**
 - O que fiz ontem que ajudou o time de desenvolvimento a atingir a meta do Sprint?
 - O que farei hoje para ajudar o time de desenvolvimento a atingir a meta do Sprint?
 - Vejo algum impedimento que me impeça, ou o time de desenvolvimento, de atender o objetivo do Sprint?

Sprint Review

- O Sprint Review é realizado no final do Sprint para inspecionar o Incremento e adaptar o **Product Backlog**, se necessário.
- Durante o **Sprint Review**, a equipe Scrum e as partes interessadas colaboram sobre **o que foi feito no Sprint**.
- Esta é uma reunião informal, não uma reunião de status, e a apresentação **do Incremento é destinado a obter feedback e promover a colaboração**.
- Esta é, no máximo, **uma reunião de quatro horas** para Sprints de um mês.
- O Scrum Master garante que o evento ocorra e que os participantes entendam seu propósito.

Sprint Review

A Revisão do Sprint inclui os seguintes elementos:

- Os participantes incluem a equipe Scrum e as principais partes interessadas, convidadas pelo PO;
- O PO explica quais itens do **Backlog do produto** foram “**Concluídos**” e o que não foi “Concluído”;
- A equipe de desenvolvimento discute o **que correu bem durante o Sprint**, quais problemas ocorreram e como foram resolvidos;
- **O PO discute o Backlog do produto**, são projetadas prováveis datas de entrega e metas com base no progresso.
- Podem ocorrer revisão do cronograma, orçamento, recursos potenciais e definição das próximas funcionalidade/incrementos.

Sprint Review

O resultado do Sprint Review é o **Product Backlog** revisado que define os itens de backlog para o próximo Sprint.

O **Product Backlog** também pode ser ajustado globalmente para atender às novas oportunidades.

Increment

O incremento é a soma de todos os itens do Product Backlog concluídos durante um Sprint, acrescido dos incrementos de todos os Sprints anteriores.

No final de um Sprint, o novo incremento deve adicionado com o status de "Concluído", o que significa que deve estar em condições de uso e atender à definição de critérios de aceite.

O incremento deve estar em condição utilizável, independentemente de o PO decidir liberá-lo ou não.

Um incremento é um conjunto de trabalhos executáveis e inspecionáveis, gerados pela Equipe de Desenvolvimento.

Sprint Retrospective

- O Sprint Retrospective é uma oportunidade para a Equipe Scrum se inspecionar e criar um plano **para melhorias a serem promulgadas durante o próximo Sprint.**
- A Retrospectiva do Sprint ocorre após o Sprint Review e antes do próximo Planejamento do Sprint.
- **Esta é uma reunião de no máximo três horas para Sprints de um mês.**
- O Scrum Master garante que o evento ocorra e que os participantes entendam seu propósito.

Sprint Retrospective

O objetivo da Retrospectiva do Sprint é:

- Inspecionar como foi o último Sprint em relação as pessoas da Equipe Scrum: relacionamentos, processos e ferramentas;
- Identificar e ordenar os principais itens que bem sucedidos, e as possíveis melhorias;
- Criar um plano para implementar melhorias no trabalho da Equipe Scrum.

Sprint Retrospective

Durante cada **Sprint Retrospective**, a Equipe Scrum planeja maneiras de aumentar a qualidade do produto, melhorando os processos de trabalho ou adaptando a definição de "Concluído".

Cancelamento do Sprint

O Sprint pode ser cancelado antes do seu período de término. Somente o PO tem autoridade para cancelar o Sprint, embora ele possa fazê-lo sob influência das partes interessadas (o Time de Desenvolvimento ou o Scrum Master, por exemplo).

User Story

User Story no Scrum

Scrum não diz como organizar e/ou documentar as tarefas que devem ser realizadas a cada Sprint, o mais comum, adotado atualmente, é através de Histórias do Usuário.

User Story é um artefato ágil para especificar requisitos.

User Story

- Uma **História do usuário** pode ser caracterizada como uma curta e simples descrição da necessidade do cliente.
- Escrita pela perspectiva do usuário, cliente do sistema ou representante de negócios do cliente.

User Story

Uma **estória de usuário** deve explicar bem para quem, o que e por que está sendo criada.

<<Título da História>>

Como um <<ator>>

Eu <<quero/preciso/desejo>> <<objetivo da história>>

Para <<razão da história ser necessária>>

Como Criar User Story

- 1) Para Quem** - Toda User Story, começa com um usuário. Este usuário é o ator principal da história, uma persona que todos no time conhecem e entendem como pensa.
- 2) O Que** – Funcionalidade não técnica, “o quê” deve ser feito.
- 3) Por que** – Motivação, justifica o motivo de criação da User Story

O Product Owner é responsável pela User Story

Dicas - User Story

- Usar o Product Backlog para criar as User Stories
- Toda funcionalidade deve impactar a vida de no mínimo uma pessoa. Se não impacta a vida de ninguém, não deveria existir.
- É importante mapear as personas que serão utilizadas antes de escrever User Stories.
- A funcionalidade deve ser descrita a nível de usuário e a nível de negócio, jamais a nível de programação.
- User Story pode possuir regras de negócio e requisitos não-funcionais.
- User Story devem possuir critérios de aceite (Teste de Aceitação).
- As User Stories abrangentes são denominadas épicas. Os épicos devem ser quebrados em histórias.

Exemplos - User Story

- Épico

Formas de Pagamento

Como um Cliente,
Eu quero que sejam disponibilizadas
diversas formas de pagamento
Para pagar meu pedido.

Fonte: <http://www.metodoti.com/2019/04/voce-ja-utilizou-historias-de-usuario%EF%BB%BF>

Exemplos - User Story

Pagamento - Boleto

Como um Cliente, quero utilizar a forma de pagamento Boleto Bancário Para pagar meus pedidos.

Pagamento - Débito em Conta - Banco 1

Como um Cliente, quero utilizar a forma de pagamento Débito em Conta do Banco 1 Para pagar meus pedidos.

Pagamento - Débito em Conta - Banco 2

Como um Cliente, quero utilizar a forma de pagamento Débito em Conta do Banco 2 Para pagar meus pedidos.

Pagamento - Cartão de Crédito - Bandeira A

Como um Cliente, quero utilizar a forma de pagamento Cartão de Crédito da Bandeira A Para pagar meus pedidos.

Pagamento - Cartão de Crédito - Bandeira B

Como um Cliente, quero utilizar a forma de pagamento Cartão de Crédito da Bandeira B Para pagar meus pedidos.

Pagamento - Cartão de Crédito - Bandeira C

Como um Cliente, quero utilizar a forma de pagamento Cartão de Crédito da Bandeira C Para pagar meus pedidos.

Fonte: <http://www.metodoti.com/2019/04/voce-ja-utilizou-historias-de-usuario%EF%BB%BF>

Exemplos - User Story

- User Story e Critérios de Aceite

Pagamento – Cartão de Crédito – Bandeira A

Como um Cliente,
Quero utilizar a forma de pagamento Cartão de Crédito
da Bandeira A
Para pagar meus pedidos.

Critérios de aceite

- Verificar que o cliente selecionou a opção de pagamento Cartão de Crédito para pagar suas compras.
- Verificar que o cliente informou um número de cartão de crédito válido (Número do Cartão, Nome, Código de Segurança, CPF).
- Verificar que o número de cartão informado é da Bandeira A.
- Verificar com a Operadora de Cartão de Crédito se existe alguma restrição que impede a utilização do cartão (extravio, furto, limite de crédito, etc).
- Estando tudo correto, prosseguir com a finalização do pedido.

Fonte: <http://www.metodoti.com/2019/04/voce-ja-utilizou-historias-de-usuario%EF%BB%BF>

Referências

- Caroli, Paulo. **Lean Inception**: Como Alinhar Pessoas e Construir o Produto. Editora Caroli. Edição do Kindle
- COHN, Mike. **Desenvolvimento de Software com Scrum**: aplicando métodos ágeis com sucesso. Porto Alegre: Bookman, 2011.
- DUARTE, Luiz. **Scrum e Métodos Ágeis**: Um Guia Prático . LuizTools. Edição do Kindle.
- RUBIN, Kenneth, **Essential Scrum**: A Practical Guide to the Most Popular Agile Process, Pearson Education, 2013.
- SCHWABER, Ken, SUTHERLAND, Jeff, **The Scrum Guide™** The Definitive Guide to Scrum: The Rules of the Game, November/2017, Disponível em
<<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>>
Acesso em 04 de fevereiro de 2020.
- SOCHOVA, Zuzana. 2017. **The Great ScrumMaster**: #ScrumMasterWay (1st. ed.). Addison-Wesley Professional.