



Laboratório de Sistemas Computacionais Complexos

2020/04/13 - AULA 10

<https://uclab.xyz/sistemas-complexos-aula10>

Renato Cordeiro Ferreira
renatocf@ime.usp.br

Thatiane de Oliveira Rosa
thatiane@ime.usp.br

João Francisco Daniel
joaofran@ime.usp.br

Alfredo Goldman
gold@ime.usp.br

Agenda

Tema da aula:

Conceitos Gerais de Desenvolvimento Ágil

1. Engenharia de Software
2. Valores e princípios ágeis
3. eXtreme Programming - XP
4. Scrum
5. Kanban
6. Modelo Spotify Squads

Engenharia de Software

Prover métodos (i.e.: **“como fazer”**), ferramentas e procedimentos (i.e.: **sequência em que os métodos serão aplicados**) para o desenvolvimento de software de maneira que ele **atinja** seus **objetivos** dentro de contextos de **sistemas complexos**

Engenharia de Software

● De forma geral, um software deve atender:

Funcionalidades;

Custo;

Tempo;

Desempenho;

Padrões;

Leis

Stakeholders...

Engenharia de Software

- **Processo** é uma **sequência de passos** que devem ser seguidos para se **atingir** determinado **objetivo**
- **Processo de desenvolvimento de software**, visa a construção de um software;
Sub processos: requisitos, análise, desenho, implementação e teste.
- Modelos de processo:
Tradicionais (cascata, espiral, prototipagem evolutiva...)
Ágeis (XP, Scrum, FDD...)

Manifesto Ágil - Valores

- **Indivíduos e interações** são mais importantes que processos e ferramentas
- **Software funcionando** é mais importante que documentação abrangente
- **Colaboração do cliente** é mais importante que negociações de contrato
- **Responder às mudanças** é mais importante que de seguir um plano

Agilemanifesto, 2001

Manifesto Ágil - Valores

- **Indivíduos e interações** são mais importantes que processos e ferramentas



Manifesto Ágil - Valores

- **Software funcionando** é mais importante que documentação abrangente

- “Telefone sem fio”



- “Documento sem fio”



Manifesto Ágil - Valores

- **Colaboração do cliente** é mais importante que negociações de contrato



Manifesto Ágil - Valores

- **Responder às mudanças** é mais importante que de seguir um plano



Manifesto Ágil - Princípios

Satisfazer o cliente por meio da entrega contínua e rápida de versões prontas do software, com valor agregado

Aceitar bem alterações de requisitos, mesmo que ocorram de forma tardia. A mudança dentro do processo ágil é entendida como parte da vantagem competitiva

Realizar entregas do software de forma frequente, a cada semana ou cada mês, com o menor tempo possível de intervalo

Possibilitar que clientes e desenvolvedores trabalhem juntos e diariamente no projeto

Construir projetos com indivíduos motivados: oferecer à equipe o ambiente e o apoio necessários, assim como confiar na capacidade da mesma em realizar o trabalho

Conversar cara a cara, esse é o método mais eficiente e eficaz para transmitir informações dentro de uma equipe de desenvolvimento

Manter o software funcionando, essa é a medida principal de progresso

Promover o desenvolvimento sustentável: clientes, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente

Manter atenção contínua à excelência técnica, pois essa prática possibilita aumentar a agilidade e o bom desempenho

Ser simples, a simplicidade é essencial para o desenvolvimento do trabalho

Possuir equipe auto-organizada, as melhores arquiteturas, requisitos e projetos, emergem de equipes que se auto-organizam

Estabelecer intervalos regulares de reflexão. A equipe deve refletir formas de se tornar mais eficiente e, em seguida, deve ajustar e aperfeiçoar o seu comportamento

Agilemanifesto, 2001

eXtreme Programming - XP

Metodologia **leve**, **eficiente**, de **baixo risco**, **flexível**,
previsível, **científica** e **divertida** (BECK, 2008)

eXtreme Programming - XP

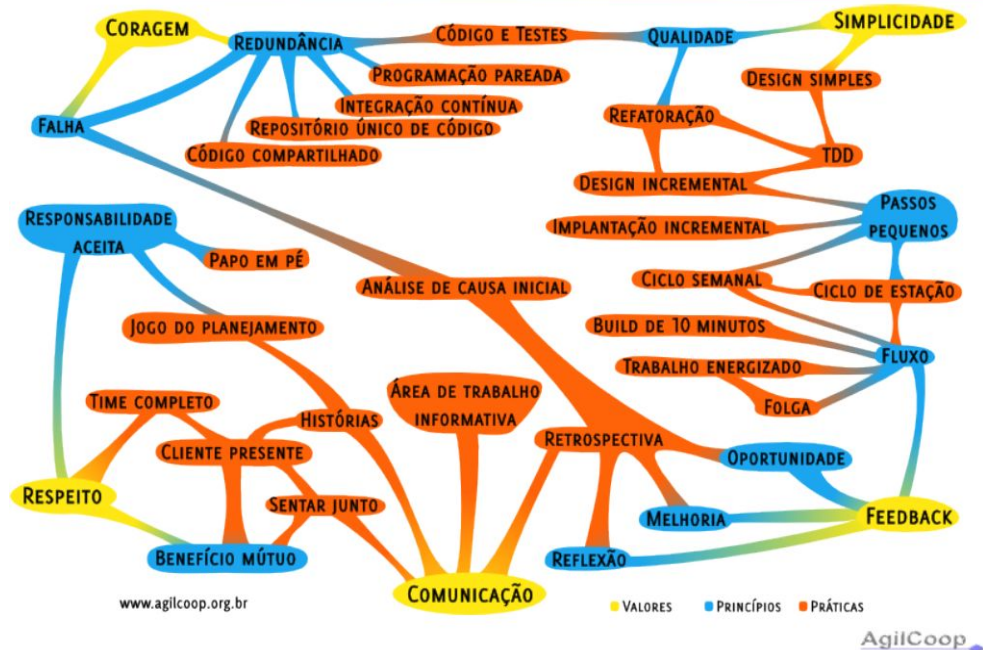
- Indicada para:

 - Equipes pequenas ou médias

 - Projetos com requisitos vagos, que modificam rapidamente

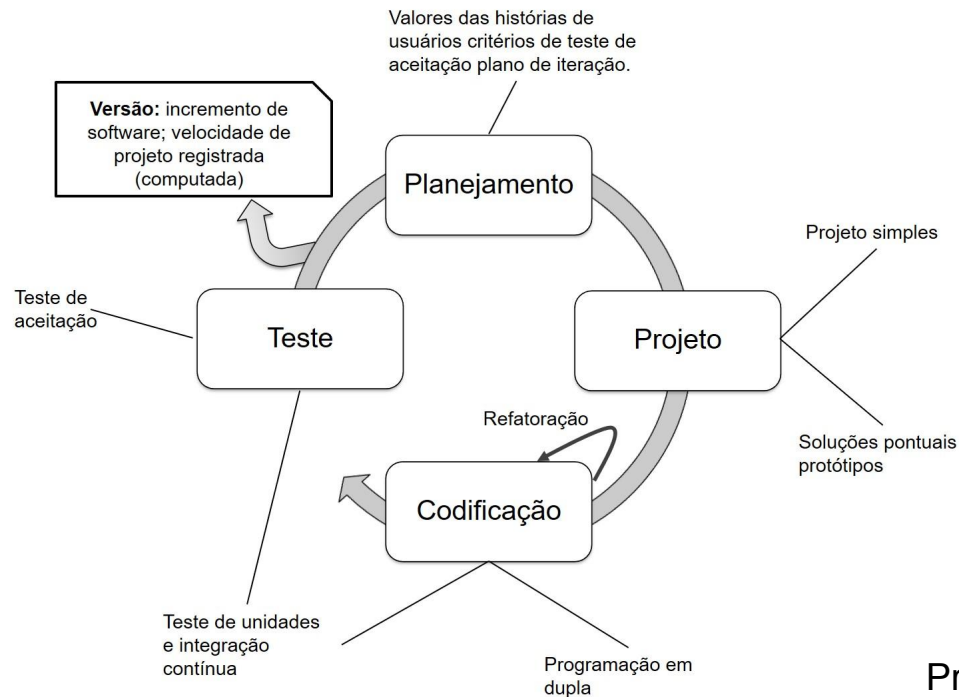
- Enfatiza do desenvolvimento do projeto de forma **rápida, simples** e que atenda as **necessidades do cliente com qualidade**, dentro do **prazo estabelecido**
- Baseia-se em um conjunto de **valores, princípios e práticas**

XP - Valores, princípios e práticas



Fonte: <http://www.agilcoop.org.br/sites/agilcoop.org.br/files//cartaz.pdf>

XP - Ciclo de vida

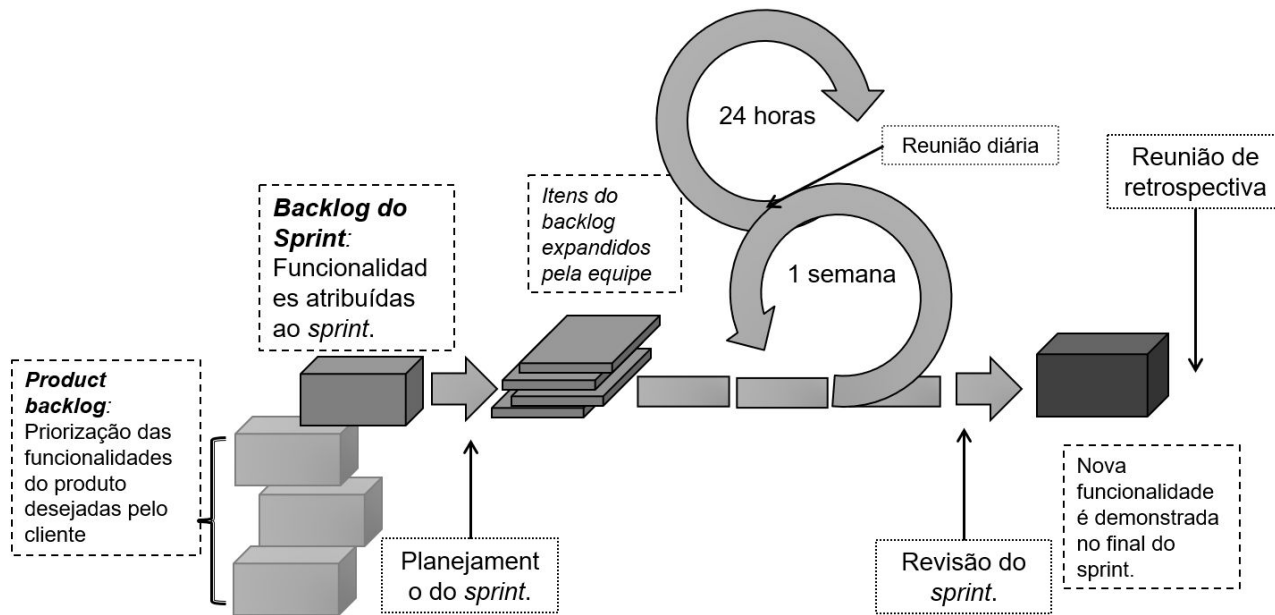


Pressman, 2011

Scrum

- Metodologia que pode ser adotada para **planejamento**, **gerenciamento** e **desenvolvimento** de produtos de software
- Tem o objetivo de controlar os **processos empíricos**, mantendo o **foco na entrega** de produtos que **agreguem valor ao negócio**, no **menor tempo possível**

Scrum - Ciclo de vida



Pressman, 2011

Kanban

- **Estratégia simples de gestão**, que utiliza quadro ou mural (físico ou digital) com “cartões” que representam atividades, que seguem um fluxo pré-estabelecido de estágios
- Os cartões mudam de estágio à medida que a execução da atividade evolui
- Se uma nova atividade é identificada, um novo cartão é criado

Kanban - Características

- Ilustra o fluxo de trabalho
- Limita a quantidade de trabalho em andamento
- Facilita a medição e otimização do fluxo de trabalho
- Torna explícitas as políticas do processo
- Facilita o gerenciamento quantitativo

Kanban

- Normalmente o quadro é dividido em três estágios:

A fazer (To Do): são tarefas planejadas para o projeto

Fazendo (Work in Progress): são tarefas que estão sendo executadas

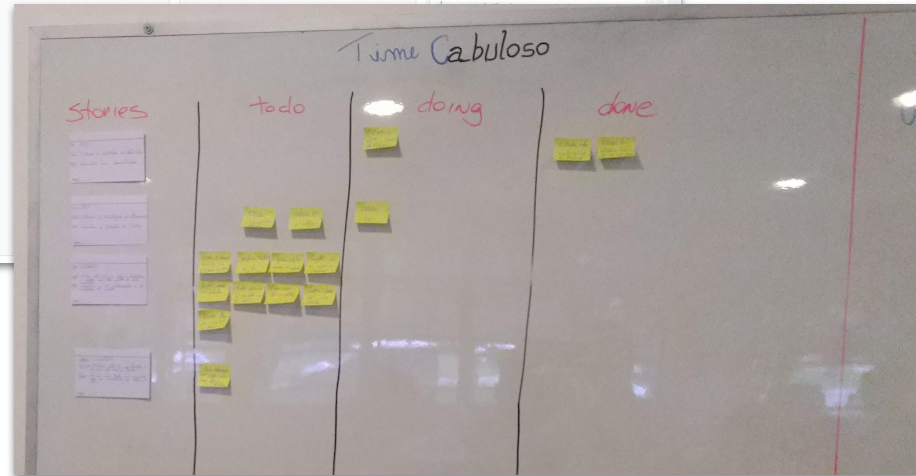
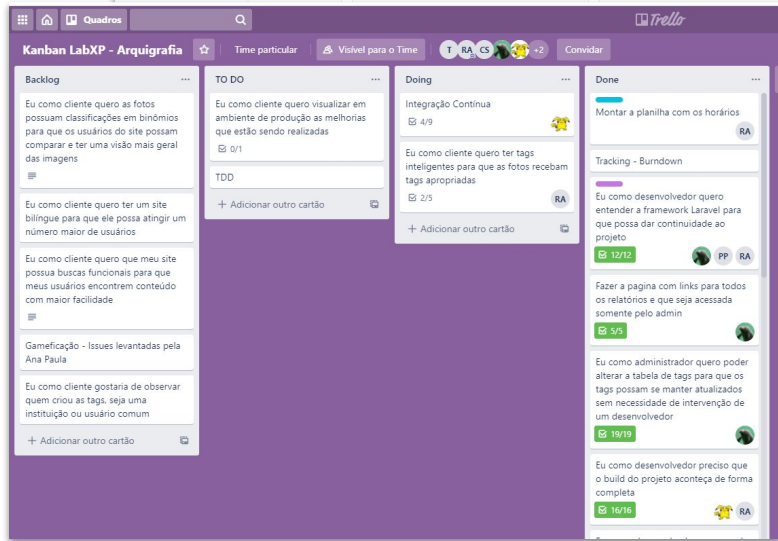
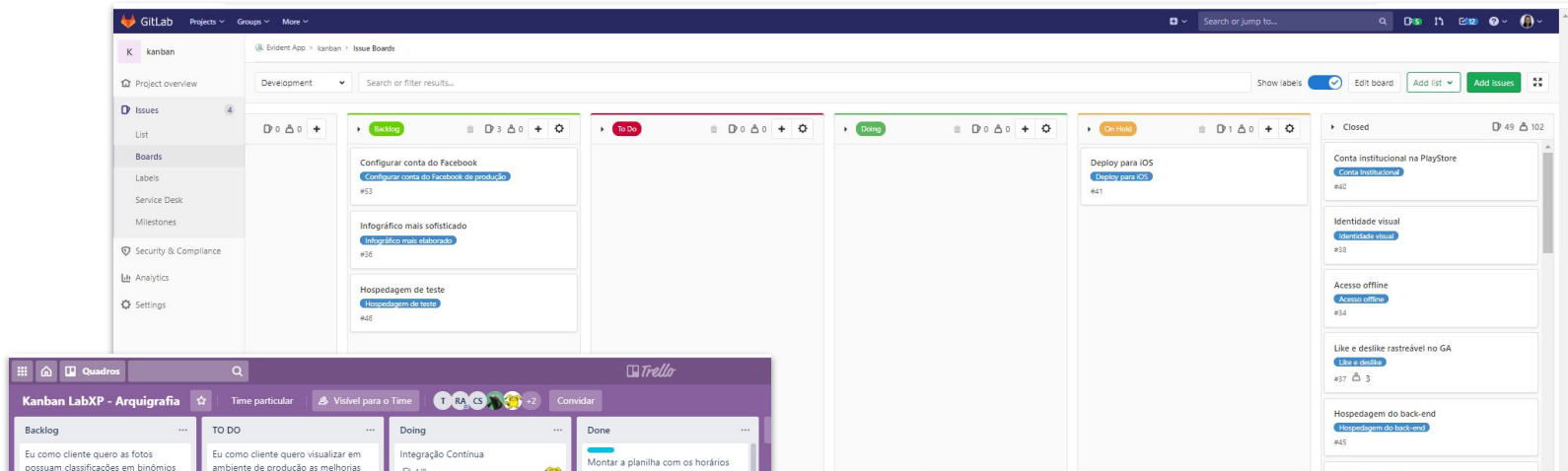
Em espera (On Hold): tarefas que foram iniciadas, mas necessitam de algum pré-requisitos para serem concluídas

Feito (Done): são tarefas já concluídas

Kanban - Histórias de usuários

- Descrições simples e curtas de funcionalidades necessárias no sistema (deve conter informações sobre o negócio)
- Estimar tempo de desenvolvimento
- Exemplo: O usuário pode querer editar as informações do seu perfil

Eu, **como** correntista
quero visualizar o meu saldo
para melhor gerenciar o meu dinheiro.



Spotify Squads

Estratégia para **escalar práticas ágeis** em **ambientes complexos**

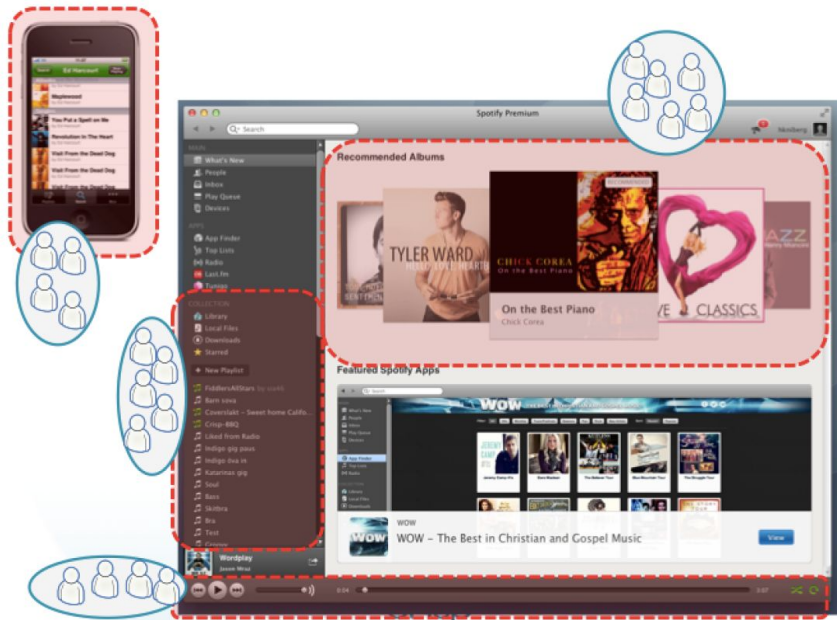
Spotify Squads - Elementos

- Squads
- Tribes
- Chapter
- Guild

Spotify Squads - Elementos - Squads

- **Equipes multidisciplinares** (desenvolvedores, PO e agile coach), **auto-organizadas** e **pequenas**
- **Executam todo o ciclo de desenvolvimento** – desde o projeto até implantação e manutenção
- Possuem **alto nível de autonomia**, decidem o que e como construir e como trabalhar

Spotify Squads - Elementos - Squads

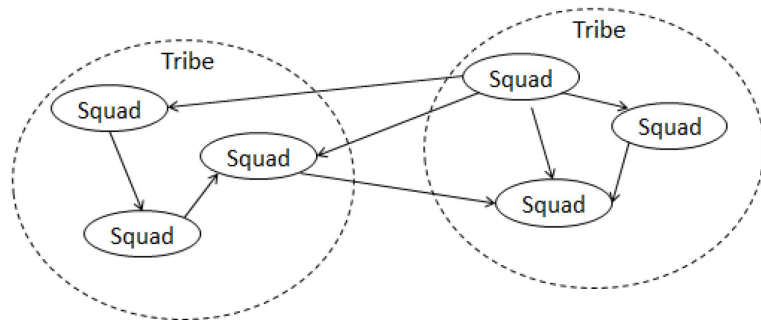


Fonte: blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf

- Cada squad é **responsável por um conjunto de funcionalidades**
- Tem **contato direto** com os **stakeholders**

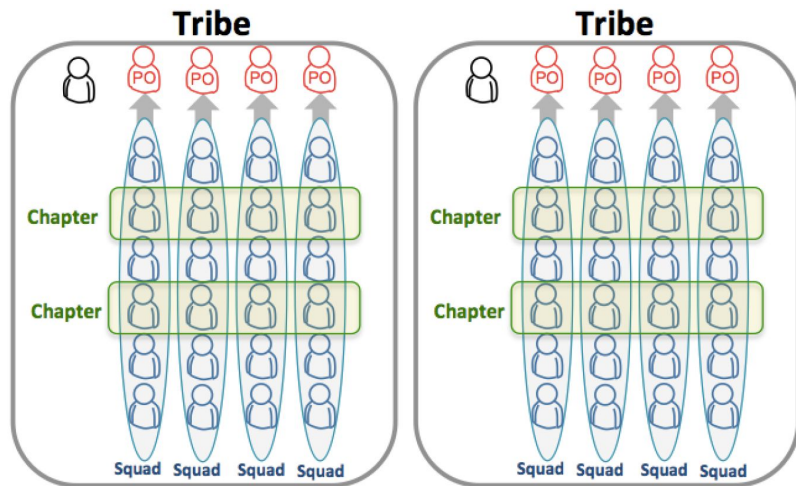
Spotify Squads - Elementos - Tribes

- **Conjunto de *squads***, que trabalham em um mesmo produto ou áreas relacionadas
- Cada *tribe* tem **total autonomia**
- **Possui um líder** para manutenção da harmonia do ambiente de trabalho



Fonte: blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf

Spotify Squads - Elementos - Chapter

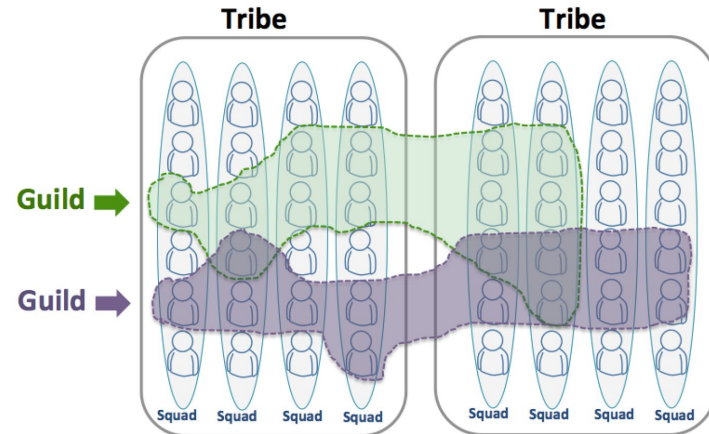


Fonte: blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf

- **Conjunto de profissionais** com as mesmas habilidades em uma área de competência, dentro da **mesma tribe**
- **Reuniões regulares**, para discutir a área de expertise e seus desafios

Spotify Squads - Elementos - Guild

- **Comunidade de interesse** que deseja **trocar conhecimento, ferramentas, códigos e práticas**
- Possui **integrantes de diferentes tribos**
- Possui um **coordenador** que ajuda a **liderar as iniciativas de cada comunidade**



Fonte: blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf

Algumas práticas e artefatos que vamos usar

- Reuniões de planejamento
- Equipe coesa
- Reuniões diárias (objetivas, focadas e rápidas)
- Design simples
- Versões pequenas
- Posse coletiva
- Programação em pares
- Padrões de codificação
- Integração contínua
- Área informativa (GitLab - Readme - Kanban)
- Reunião de retrospectiva
- Histórias de usuário
- Product backlog
- Sprint backlog

- Free Webinar: **Effective Retrospectives in the Age of Coronavirus**

- Próxima terça-feira às - 12:00, horário de Brasília

- Link para inscrição:

https://zoom.us/webinar/register/WN_vA9_ioGY_S_2d-O877EjGtQ



April 10, 2020

						
Diana Larson Co-Founder Agile Fluency Project Agile Retrospectives Expert AFP	Esther Derby President Esther Derby Associates Inc. Agile Retrospectives Expert esther derby	Molood Ceccarelli CEO Remote Forever Remote Agile Expert Remote Forever	Paul Tevis Coach and Facilitator Vigorous Consulting Remote Agile Expert Vigorous	Jay-Allen Morris Remote Team Coach Eyes Remote Facilitation Expert Eyes	Kirsten Clacey Remote Team Coach Eyes Remote Facilitation Expert Eyes	David Horowitz Co-Founder & CEO Retrium Remote Retrospectives Expert Retrium

With Support From:  Retrium 

Free Webinar
Effective Retrospectives in the Age of Coronavirus
Next Tuesday, April 14 at 11:00 AM ET

Hi there,

The world is suffering. Companies are struggling to adapt to remote work. People are overwhelmed by these unprecedented changes to the way we live, work, and connect with those around us.

Throughout this series, many of you have reached out wanting to continue the conversation around how to navigate these transitions.

That's why we've asked Agile and retrospective experts to join us next **Tuesday, April 14 at 11:00 AM ET** for the next discussion in our Suddenly Distributed series, **Effective Retrospectives in the Age of Coronavirus**.

Referências

AGILE. **Manifesto for Agile Software Development**, 2001. Disponível em: <<https://agilemanifesto.org/>>.

BECK, Kent. **Programação Extrema Explicada: Acolha as mudanças**. Porto Alegre: Bookman, 2008. 182.

BOEG, Jesper. **Kanban em 10 Passos**. 1ª Ed. São Paulo: C4Media, 2011.

KNIBERG, Henrik; IVARSSON, Anders. **Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds**, 2012. Disponível em: <<http://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf>>

KNIBERG, Henrik; SKARIN, Mattias. **Kanban e Scrum obtendo melhor de ambos**. 1ª Ed. São Paulo: C4Media, 2009.

LEFFINGWELL, Dean. **Scaling Software Agility: Best practices for large enterprises**. Boston: Person Education, 2007.

MARTIN, Robert Cecil . **Agile Software Development: principles, patterns, and practices**. New Jersey: Person Education, 2002.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 7ª ed. Porto Alegre: AMGH, 2011. 780.

WAZLAWICK, Raul Sidnei. **Engenharia de Software: Conceitos e Práticas**. Rio de Janeiro: Elsevier, 2013. 343.

Licença

Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições:

Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença.

Mais detalhes sobre essa licença em: creativecommons.org/licenses/by-nc-sa/3.0/

Créditos

Imagens usadas nesta apresentação são provenientes de: [freepik.com](https://www.freepik.com)



Laboratório de Sistemas Computacionais Complexos

2020/04/13 - AULA 10

<https://uclab.xyz/sistemas-complexos-aula10>

Renato Cordeiro Ferreira
renatocf@ime.usp.br

Thatiane de Oliveira Rosa
thatiane@ime.usp.br

João Francisco Daniel
joaofran@ime.usp.br

Alfredo Goldman
gold@ime.usp.br