

Engenharia de Software

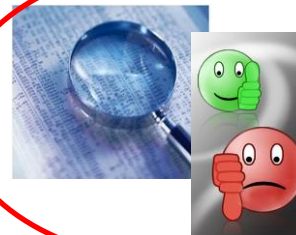
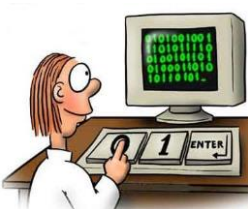
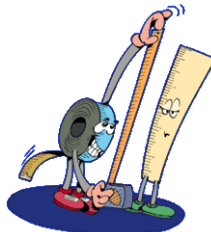
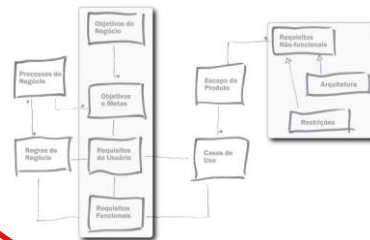
Natália Schots

Agenda

- Desenvolvimento ágil
 - Manifesto ágil
 - Lean
 - Scrum
 - XP – eXtreme Programming
 - Kanban

Na aula passada...

Processos do MR-MPS-SW



Desenvolvimento Ágil

O que é?

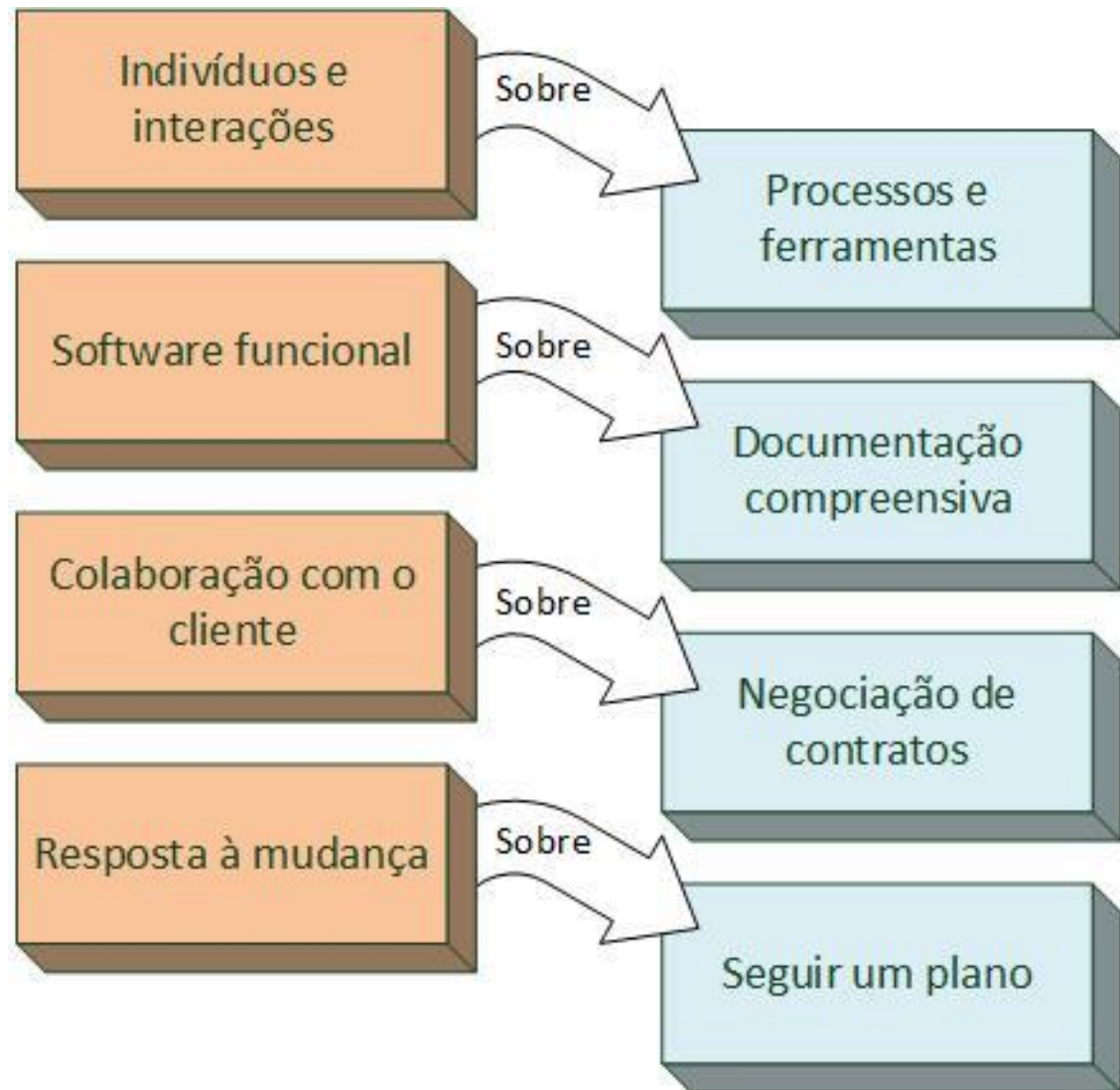
- É uma **mentalidade** (*mindset*) para o desenvolvimento de software prezando a entrega de valor para o cliente
- São dadas respostas rápidas e flexíveis a mudanças
 - O projeto é replanejado continuamente
 - São feitas entregas incrementais e constantes do software, refletindo as mudanças solicitadas

Manifesto Ágil (1/2)

- Publicado em 2001
 - <https://agilemanifesto.org/iso/ptbr/manifesto.html>
- Resultado de um encontro de desenvolvedores para discutir métodos de desenvolvimento de software mais “leves”

Manifesto Ágil (2/2)

“Mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.”



Princípios Ágeis

- Prover **satisfação** para o cliente
- Acolher **modificações** nos requisitos
- Entregar o software com **frequência**
- Trabalhar **junto** ao cliente
- Manter os indivíduos **motivados**
- Promover conversas **face a face**
- Medir o **progresso** com software funcionando
- Manter um **ritmo constante** de trabalho
- Cuidar da **qualidade**
- Buscar por **simplicidade**
- Trabalhar com equipes **auto-organizadas**
- Ajustar o comportamento da equipe buscando mais **efetividade**

Principais Métodos



Lean (1/2)

- Originalmente, criado pela Toyota como *Lean Manufacturing*
- Possui o objetivo de produzir produtos e serviços com maior qualidade, atendendo melhor o cliente
- Focado em **reduzir desperdício**
 - Entrega contínua, enxuta, no menor tempo e com qualidade

Lean (2/2)

- Princípios:
 - Elimine desperdícios
 - O que não agrega valor ou não é percebido pelo cliente
 - Inclua a qualidade no processo
 - Crie conhecimento a partir do compartilhamento
 - Adie decisões para diminuir incertezas
 - Entregue rápido
 - Respeite as pessoas e delegue poderes à equipe
 - Otimize o todo

Scrum (1/3)

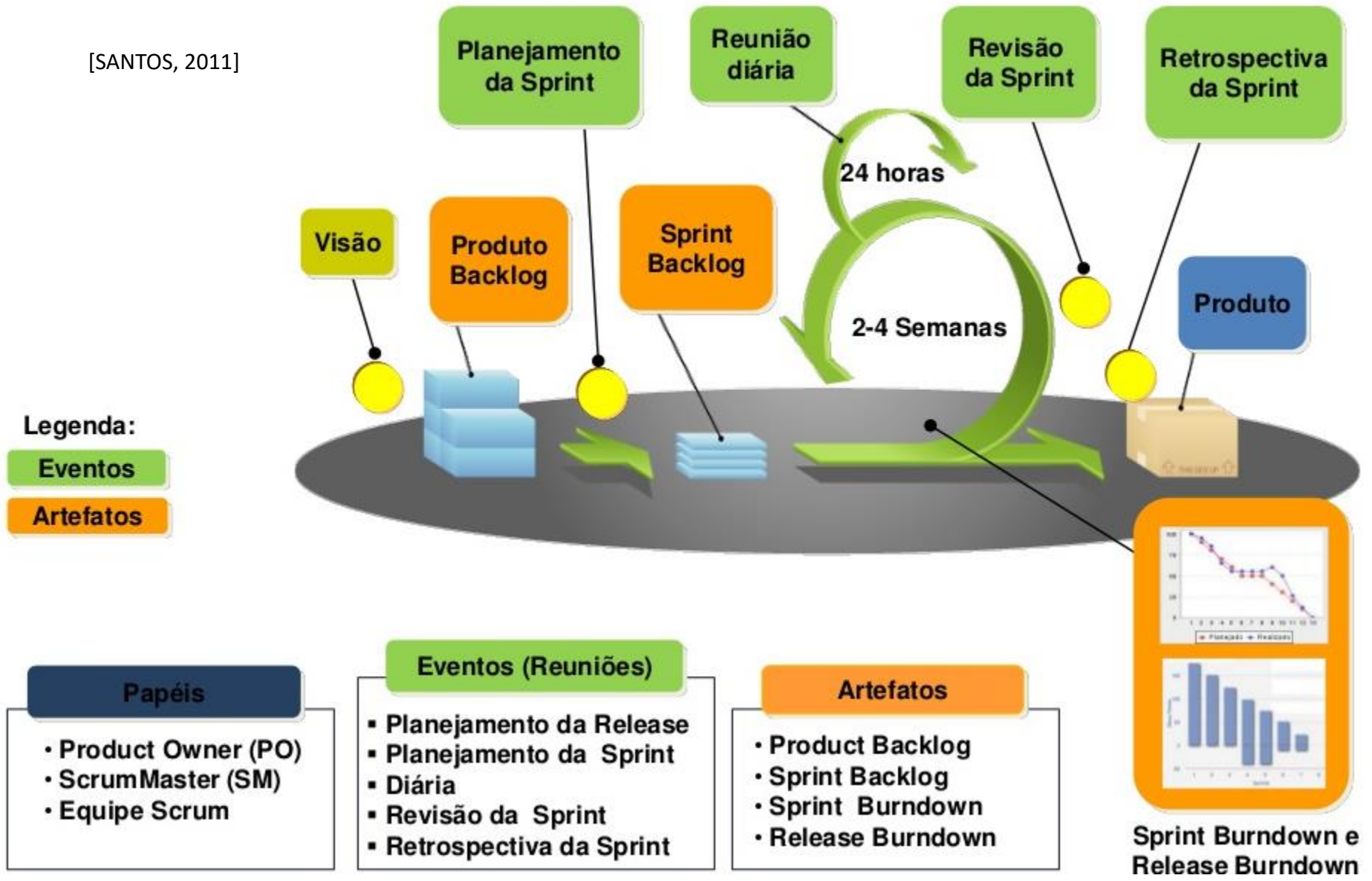
- “Um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível” (Schwaber e Sutherland)
- Guia do Scrum disponível em:
<https://www.scrumguides.org>

Scrum (2/3)

- Principais características:
 - Mais voltada para a parte **gerencial**
 - Pequenas equipes de trabalho organizadas para **maximizar comunicação e minimizar supervisão**
 - Equipes sem nível hierárquico nem papéis
 - Eventos com **duração fixa** (*time-boxes*)
 - Pilares:
 - Transparência
 - Inspeção
 - Adaptação

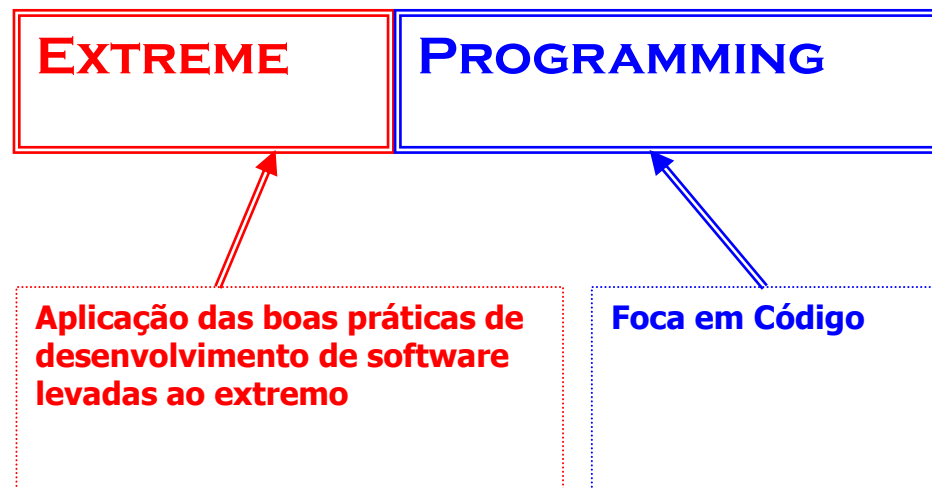
Scrum (3/3)

[SANTOS, 2011]



XP – *eXtreme Programming* (1/3)

- Método ágil bastante utilizado, desenvolvido para:
 - Equipes pequenas e médias (2 a 12 pessoas)
 - Requisitos vagos e em constante evolução

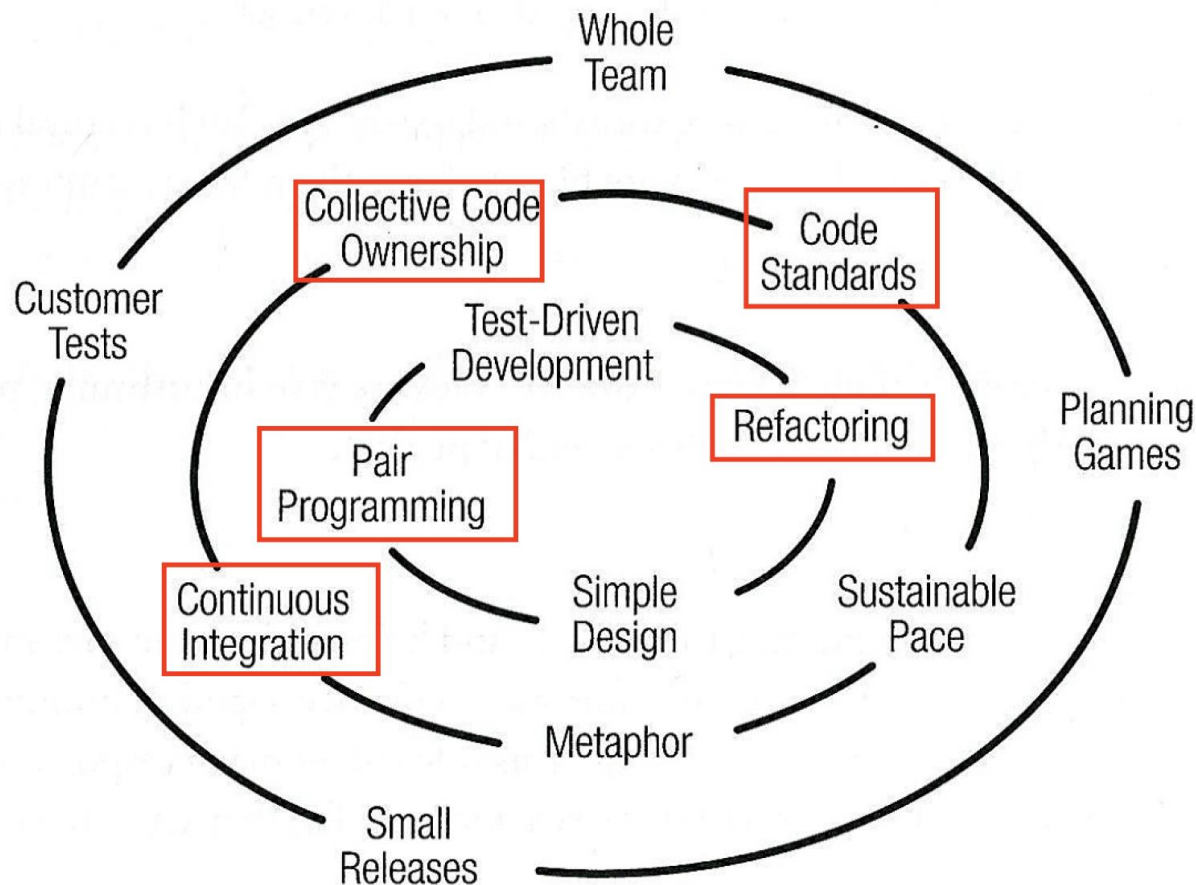


XP – *eXtreme Programming* (2/3)

- Principais características
 - Cliente disponível ou presente no dia-a-dia do projeto
 - Reuniões diárias de 15 a 20 minutos
 - Programação em par
 - Dois desenvolvedores trabalham no mesmo problema
 - Código coletivo: todos conhecem o código
 - *Releases* curtos: entregas frequentes para o cliente
 - Desenvolvimento guiado por testes (TDD)
 - Integração contínua

XP – *eXtreme Programming* (3/3)

- Principais características (cont.)



Kanban (1/4)

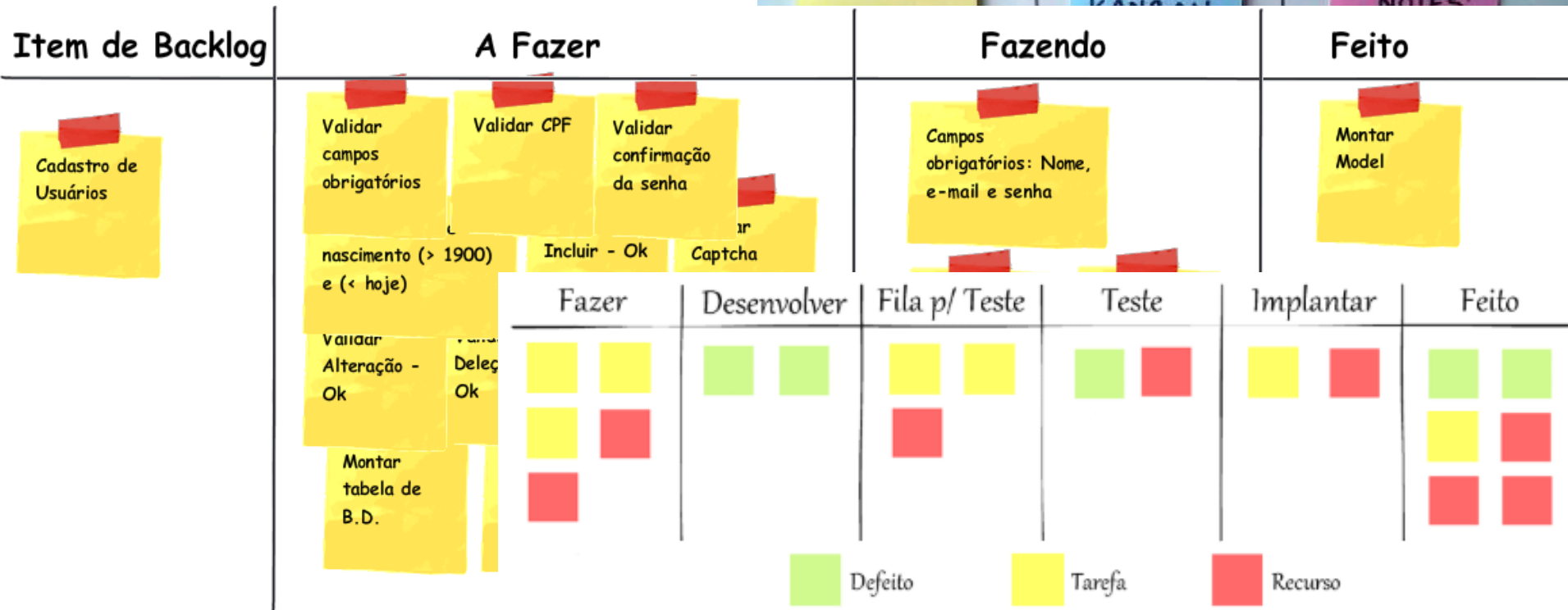
- Uma das práticas do Sistema Toyota de Produção
 - Visa auxiliar as organizações a terem um fluxo de trabalho mais eficiente e alinhado com as demandas de seus clientes
- Não é um método de desenvolvimento, apenas uma ferramenta para auxiliar na organização de tarefas

Kanban (2/4)

- Significa “sinalização visual”: *kan* (visual) + *ban* (quadro ou cartão)
- É utilizado (parcialmente) no desenvolvimento de software para otimizar o processo
 - Foco no trabalho em progresso, apresentando a evolução de forma visual (Quadro Kanban)

Kanban (3/4)

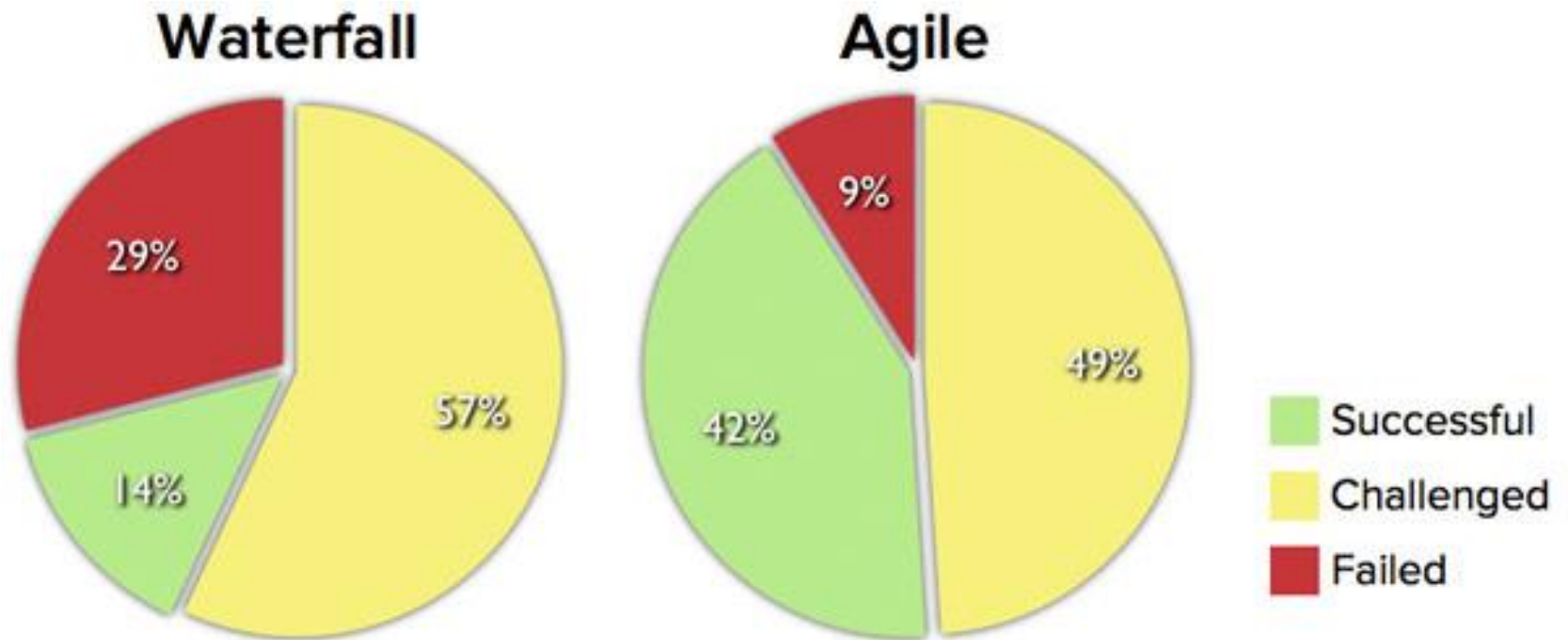
- Exemplos de Quadros Kanban



Kanban (4/4)

- Principais práticas
 - Visualizar o fluxo de trabalho (*workflow*)
 - Limitar a quantidade de trabalho em andamento (WIP - *Work in progress*)
 - Gerenciar e medir o fluxo
 - Tornar explícitas as políticas do processo
 - Implementar *loops* de *feedback*
 - Reconhecer oportunidades de melhoria

Cascata x Ágil

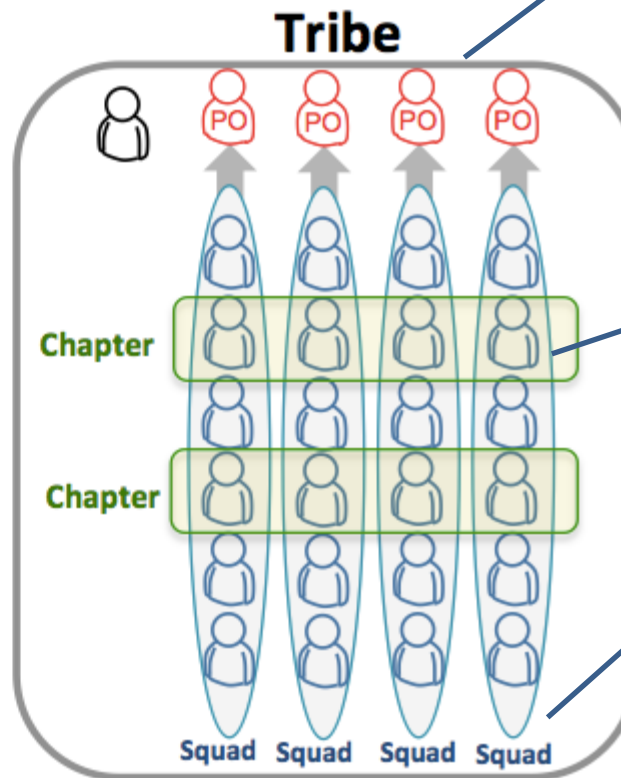
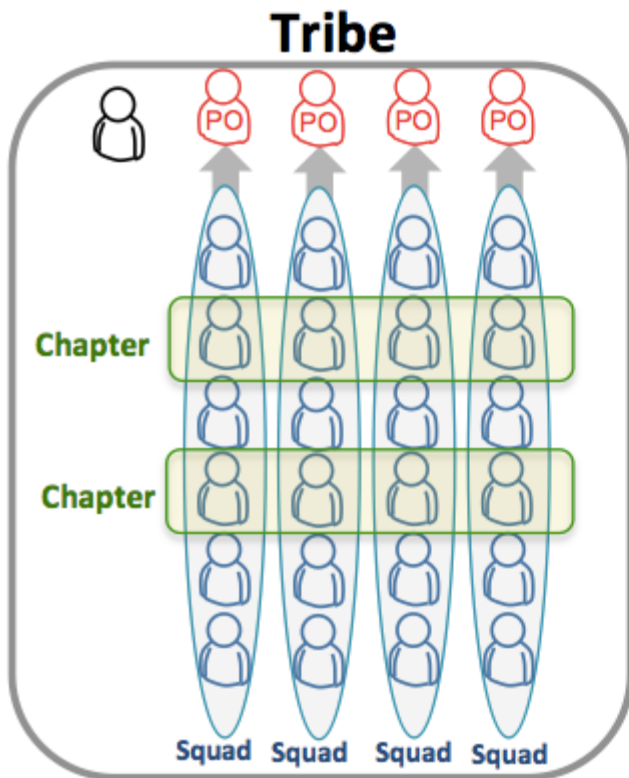


Source: The CHAOS Manifesto, The Standish Group, 2012.

Exemplo – Adaptação Ágil



- Estrutura organizacional:



Tribo: várias squads que trabalham com funcionalidades (*feature*) correlacionadas

Chapter: conjunto de profissionais que possuem especialidades (*skills*) e responsabilidades semelhantes

Squad: vários profissionais com especialidades complementares responsáveis por um conjunto de funcionalidades (*feature*)

Lembrete!

- Na próxima aula, dia 25/nov:
 - Apresentação da Etapa 1 do trabalho
 - Grupos que irão apresentar (nesta ordem)
 - Grupo 6
 - Grupo 2
 - Grupo 10
 - Grupo 12
 - Grupo 1
 - Grupo 4
 - OBS: **Todos** os grupos devem enviar os slides da apresentação no SIGAA

Referências

- Pressman, R.S.; “Engenharia de Software”; 6ª edição, Ed. McGraw-Hill, 2006
- Sommerville, I., “Engenharia de Software”, 11ª edição, Ed. Pearson, 2019
- Slides Engenharia de Software – Professor Leonardo Murta
- Slides Engenharia de Software – Professor Marcelo Schots
- Slides Processo de Software – Professora Ana Regina Rocha
- Slides Introdução à Engenharia de Software – Professor Márcio Barros
- Slides Processo de Software – Professor Gleison Santos
- Slides Métodos Ágeis - Professor Marcelo Schots
- [Silva et al., 2011] Silva, T., Magela, R., Santos, G. et al. “Implantação do Nível F do MR-MPS Combinando Características do Processo Unificado com Práticas SCRUM”, WAMPS, 2011, Campinas.