

Processo de Desenvolvimento de Software Ágil



Prof. Mauricio Seiji

www.sc.senai.br | 0800 48 1212

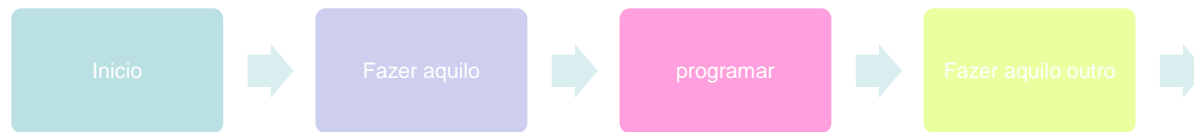
Ementa

- O que é um processo de desenvolvimento de software?
- Métodos Tradicionais Vs Métodos Ágeis
- Uma Viagem no Tempo
- Contexto da Turma
- O Manifesto Ágil
- Princípios Ágeis
- Métodos Ágeis na Prática

O que é um processo de desenvolvimento de software?



Objetivos do processo de desenvolvimento de software



- ✓ Orçamento
- ✓ Cronograma
- ✓ Usuário Feliz

Precisamos de um processo, de uma metodologia ou de uma engenharia pra construir software?



O que pode dar errado? (1994)

Entre 80 e 90% dos projetos de IT fracassam. Cerca de 30% são cancelados e em torno de 52% custam mais do que a estimativa inicial.

Fonte: www.standishgroup.com

O que pode dar errado? (2008)

- 35% dos requerimentos mudam ao longo do ciclo de vida do software
- 45% das funcionalidades entregues jamais são usadas
- 82% dos projetos de software são incompletos ou instáveis

Fonte: <http://techdistrict.kirkk.com/2008/05/01/software-development-failure/>

O que pode dar errado? (2009)

- 51% dos projetos de software pesquisados em 2009 estavam atrasados, custaram mais do que estimado e/ou foram entregues sem funcionalidades consideradas indispensáveis.

Fonte: http://www1.standishgroup.com/newsroom/chaos_2009.php

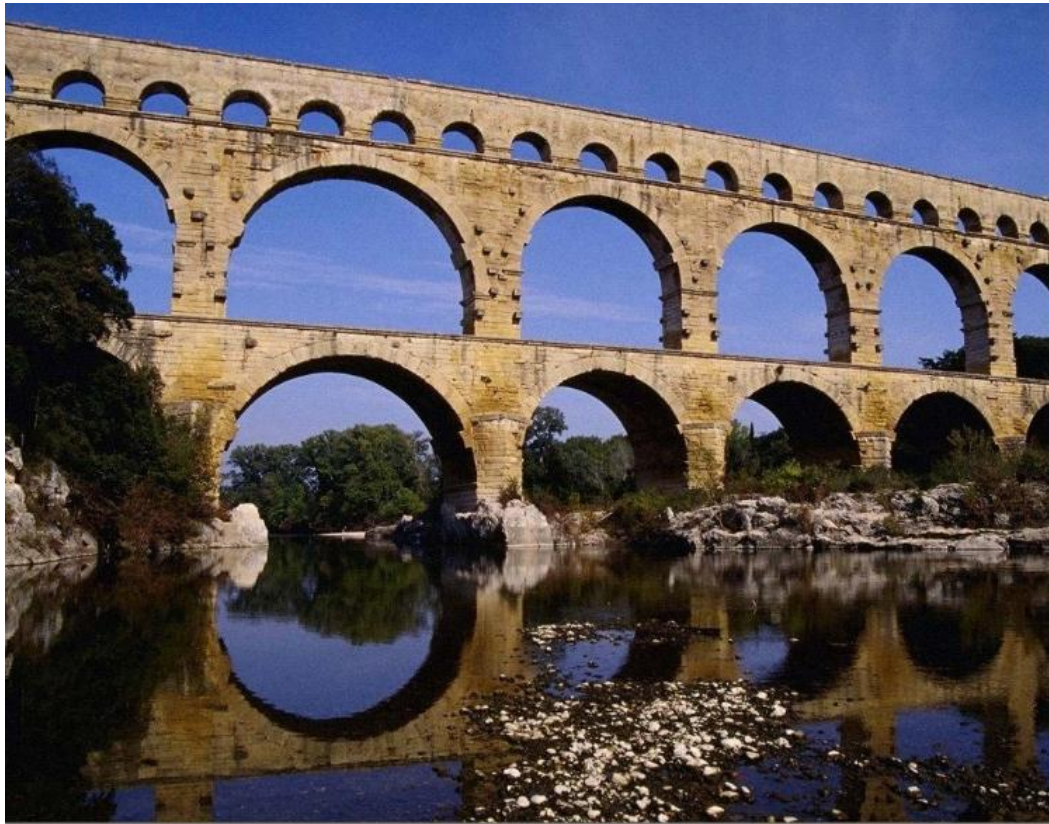
Para saber mais: http://www.it-cortex.com/Stat_Failure_Rate.htm

O que pode dar errado? (2014)

- 31% dos projetos de software pesquisados em 2014 foram cancelados antes do fim. 52.7% dos projetos tiveram custo maior que o orçamento inicial

Fonte: <http://www.projectsmart.co.uk/docs/chaos-report.pdf>

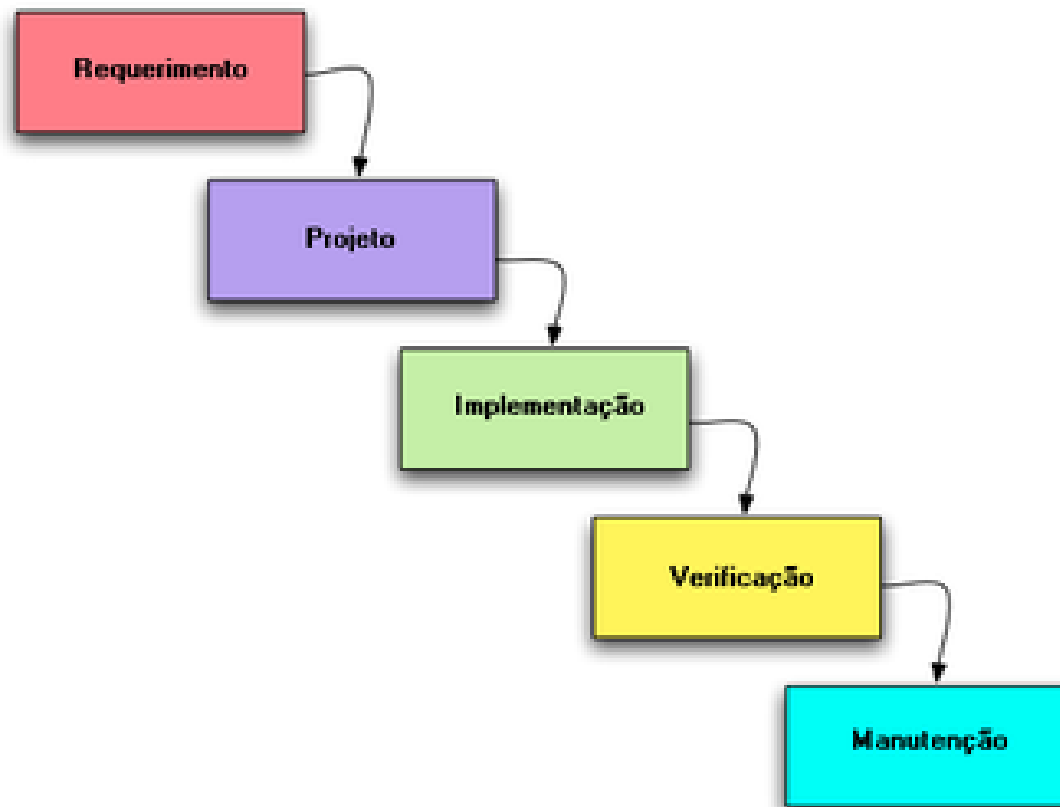
A história do desenvolvimento de software



Uma breve viagem no tempo



Exemplo de método tradicional: Sequencial Linear



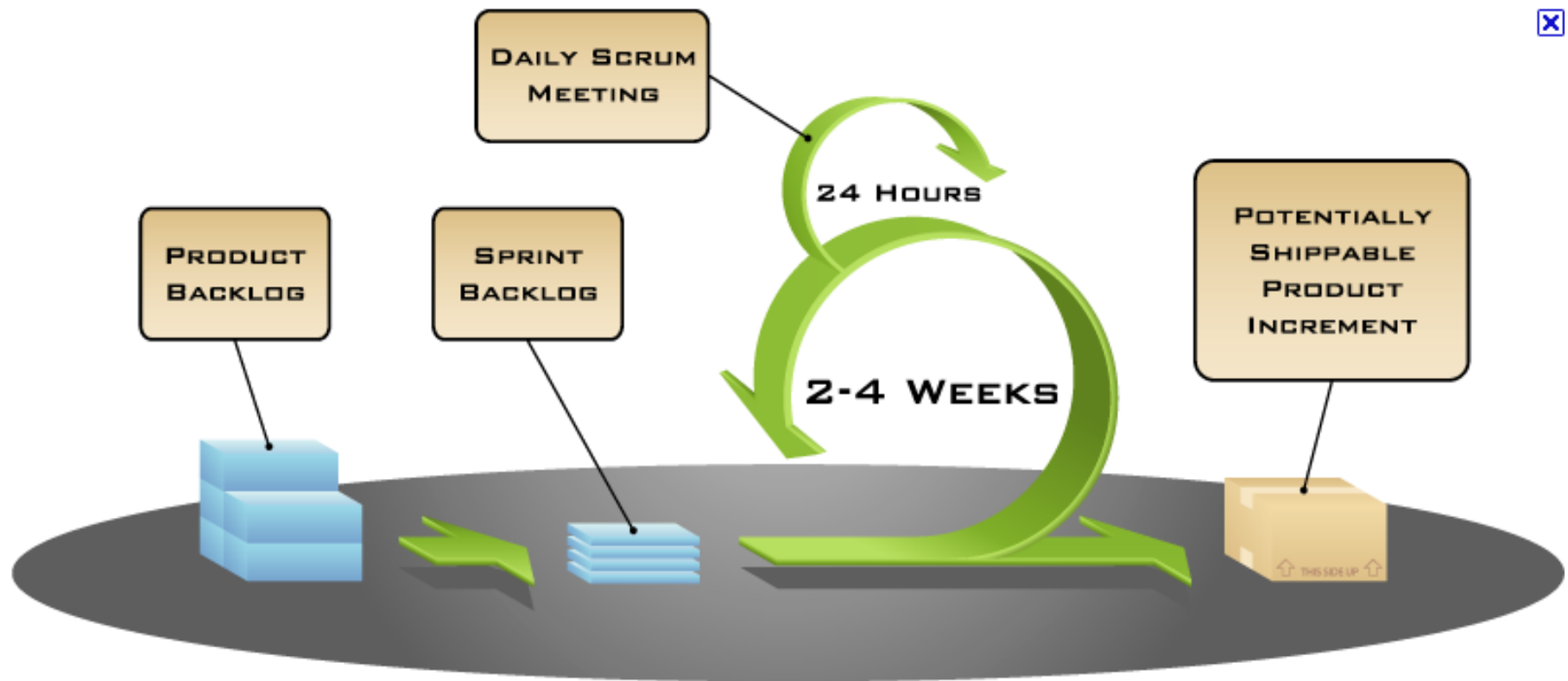
Características dos métodos tradicionais

- Abordagem preditiva
- Fundamentado em documentação abrangente
- Escopo definido em contrato
- Baixo envolvimento do cliente

Exercício (em dupla) – Como funciona o processo de desenvolvimento na empresa em que você trabalha?

- Esquema apresentando das fases do processo de desenvolvimento de software
- Vantagens desse processo de desenvolvimento
- Desvantagens

Exemplo de método ágil: Scrum



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Características dos métodos ágeis

- ~~Abordagem preditiva~~
 - Abordagem adaptativa
- ~~Fundamentado em documentação abrangente~~
 - Fundamentado em colaboração
- ~~Escopo definido em contrato~~
 - Esforço definido em contrato
- ~~Baixo envolvimento do cliente~~
 - Alto envolvimento do cliente

Manifesto ágil

- **Os indivíduos e suas interações** acima de procedimentos e ferramentas;
- **O funcionamento do software** acima de documentação abrangente;
- **A colaboração dos clientes** acima da negociação de contratos;
- **A capacidade de resposta às mudanças** acima de seguir um plano pré-estabelecido;

A metodologia ágil entende que, embora exista valor nos itens à direita, existe um valor maior nos itens à esquerda.

Princípios ágeis

- Um produto simples, mas útil a cada iteração;
- Aceitação e adaptação às mudanças;
- Rápido e freqüente “feedback” ao cliente;
- Times auto-organizados.

O que é mais importante do que a engenharia de software?

Métodos ágeis na prática

Stand up daily meeting

(A reunião diária em pé)

- 1. O que você fez desde a última reunião diária?**
- 2. O que você fará até próxima reunião diária?**
- 3. O que te impede de fazer seu trabalho da melhor maneira possível?**

*** Apenas desenvolvedores (que escrevem código fonte) podem falar.**

Principais Abordagens ágeis

- Princípios Lean
- Kanban
- Extreme Programming (XP)
- Scrum

Princípios Lean

Também é chamado de Sistema Toyota ou de Produção Enxuta.

“Lean é um conjunto de princípios e não um processo de desenvolvimento de software.”

Fonte: <http://leanandkanban.files.wordpress.com/2009/04/kanban-for-software-engineering-apr-242.pdf>

Lean – Redução do desperdício



Os 5 princípios do Lean thinking



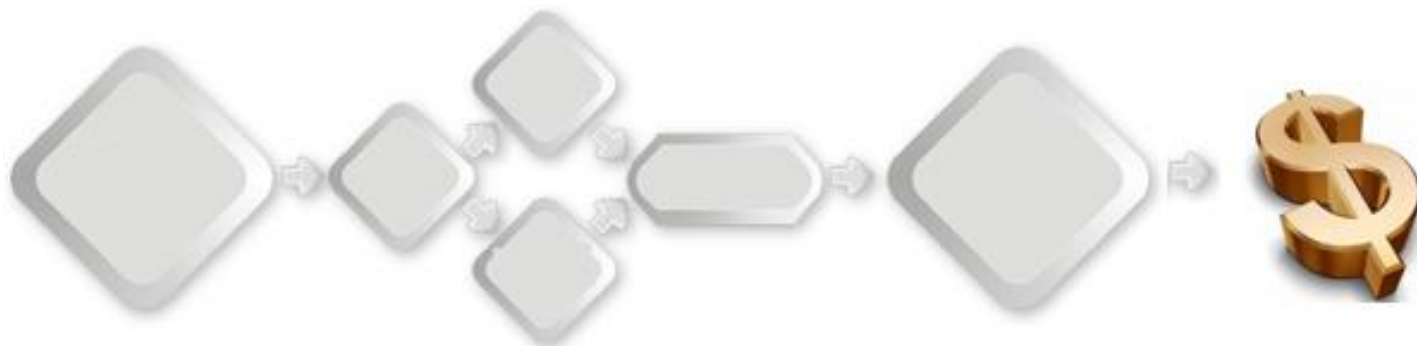
Fonte: <http://www.lean.org.br>

Os 5 princípios do Lean thinking Valor



Os 5 princípios do Lean thinking

Fluxo de Valor



Os 5 princípios do Lean thinking

Fluxo Contínuo



Os 5 princípios do Lean thinking

Produção Puxada (pelo cliente)



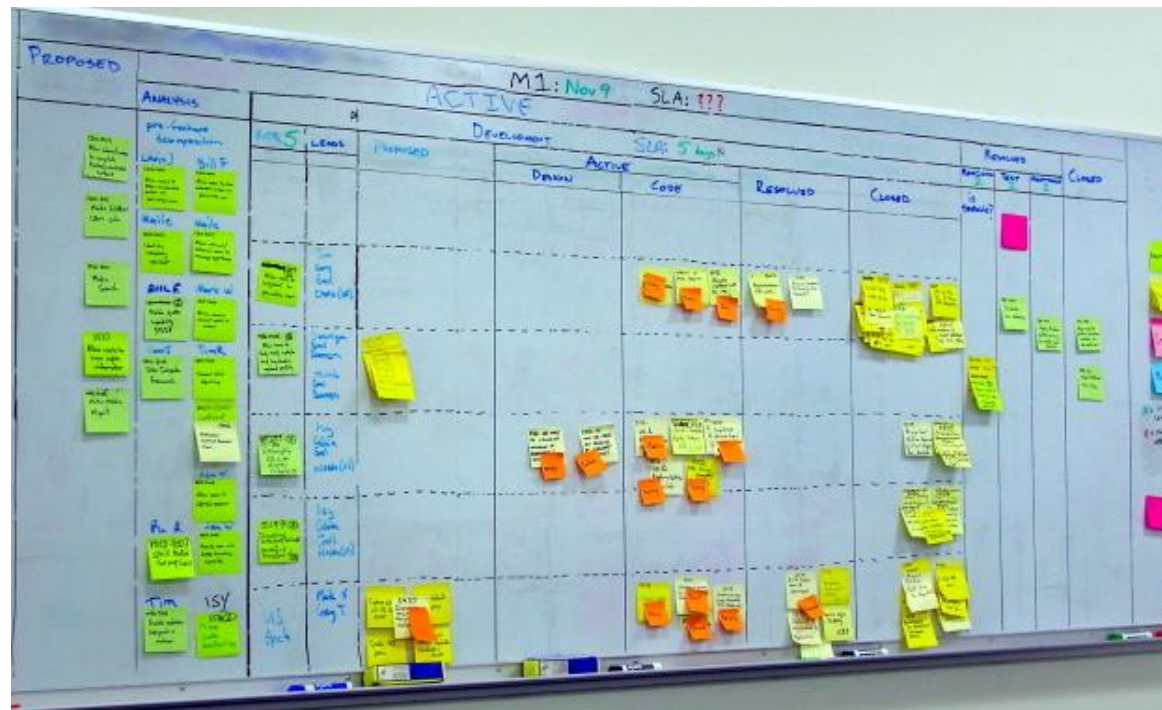
Os 5 princípios do Lean thinking

Perfeição



Kanban

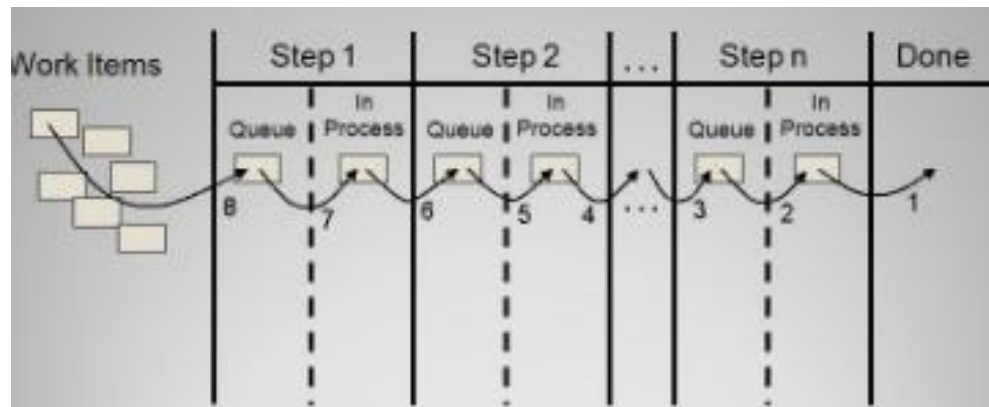
Kanban é uma palavra japonesa que significa literalmente **registro** ou **placa visível**.



Fonte: <http://www.lean.org.br>

Kanban

Kanban is a scheduling system that tells you what to produce, when to produce it, and how much to produce.



Fonte: <http://en.wikipedia.org/wiki/Kanban>

Os princípios do Kanban Limit Work in Progress (WIP)

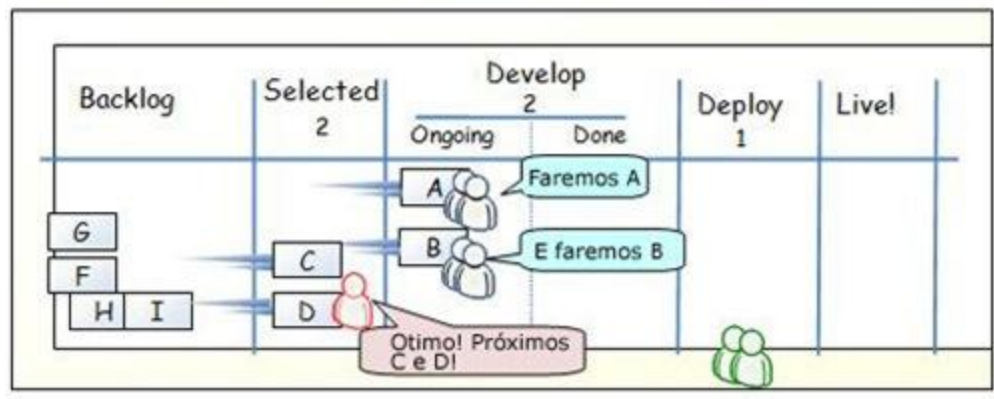
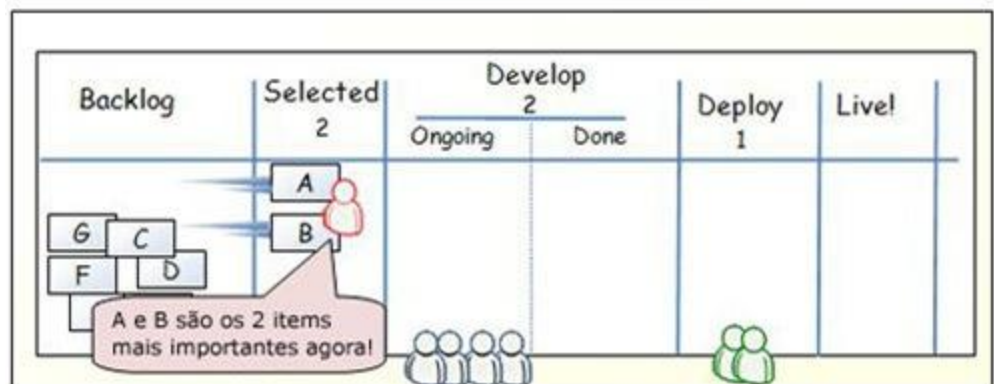
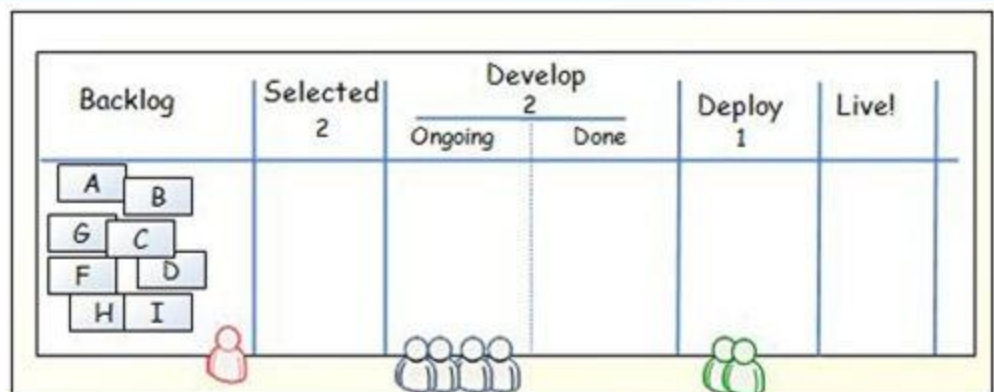


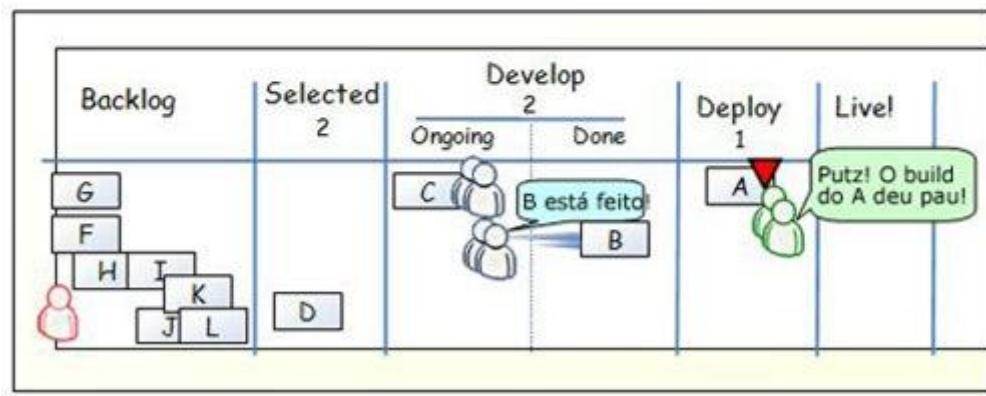
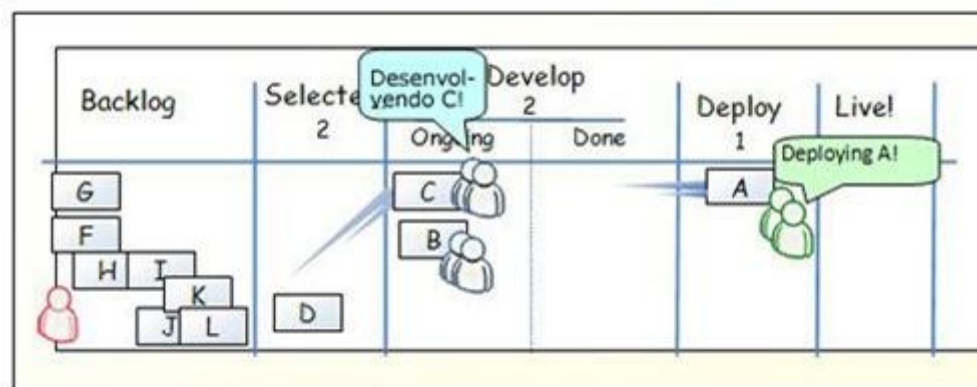
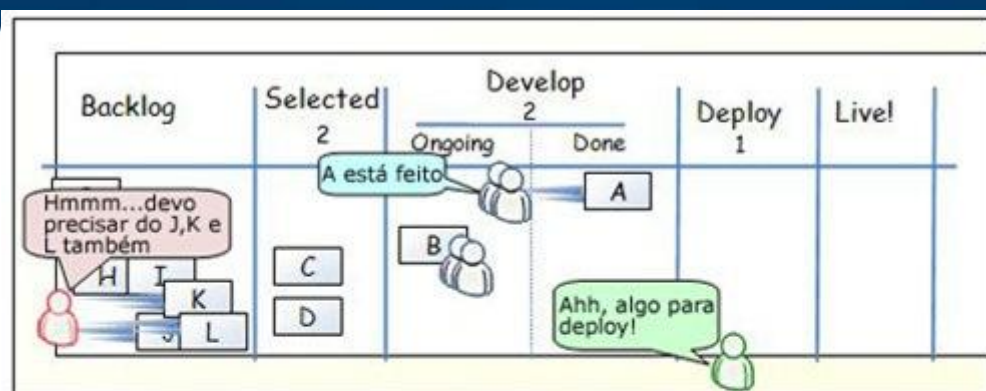
Os princípios do Kanban

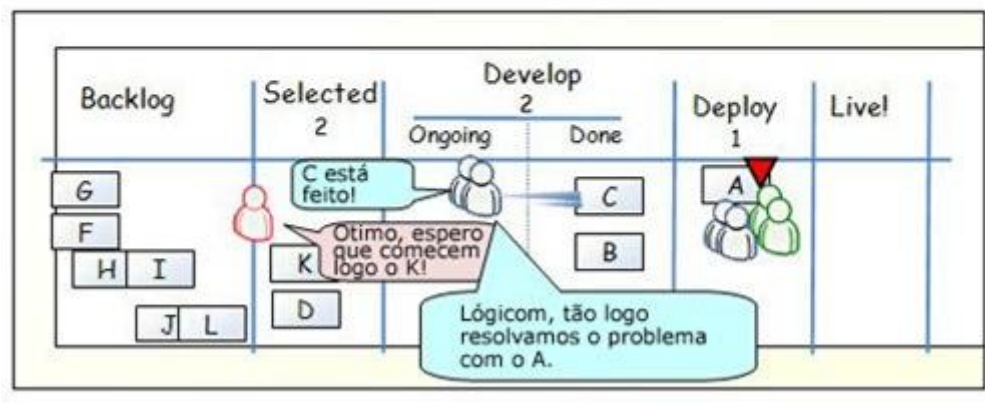
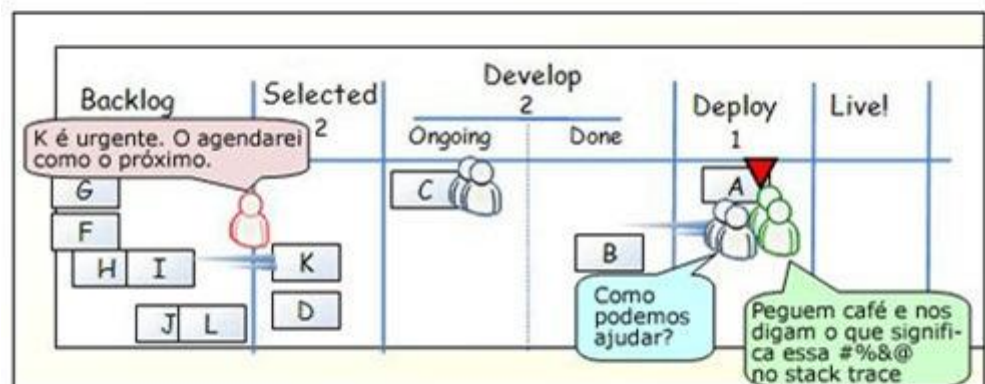
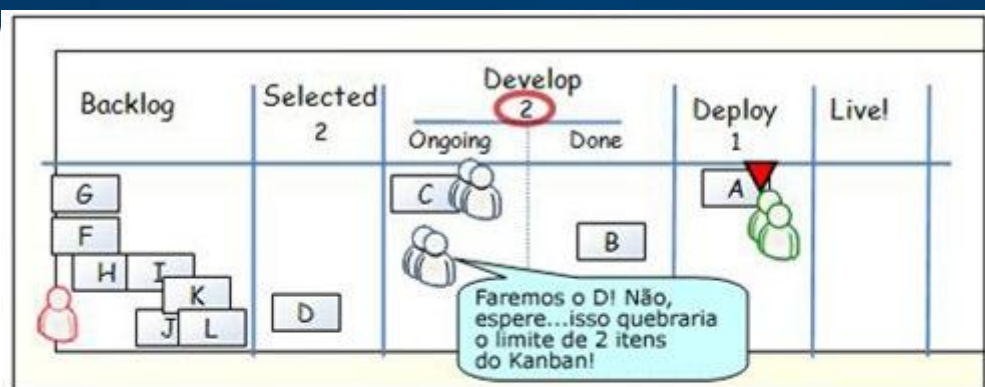
Fluxo de trabalho visível

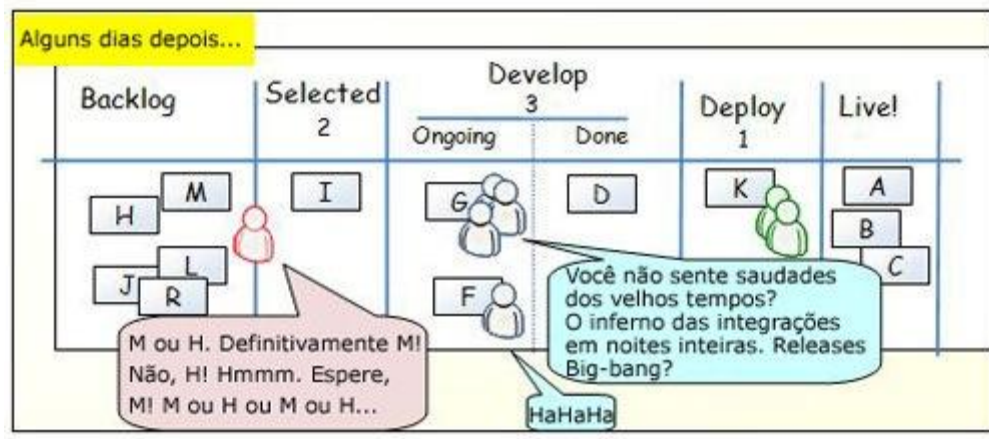
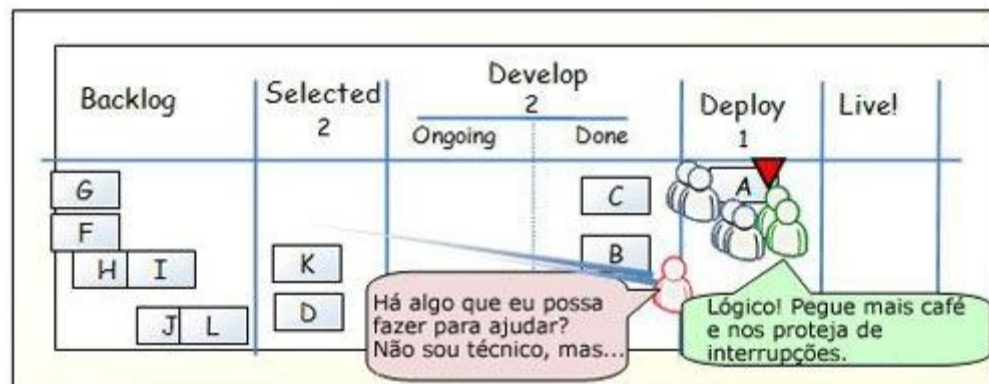
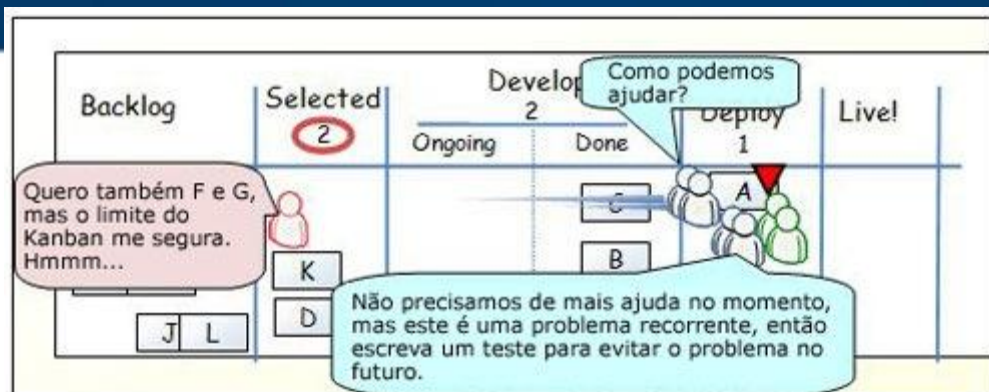
Quadrinhos

<http://www.olharcritico.com/engenhariadesoftware/2009/12/um-dia-na-terra-do-kanban/>










Por que Kanban?

- 
- Não construir funcionalidades que não são necessárias agora
 - Não escrever mais especificações do que a equipe pode codificar
 - Não escrever mais código do que a equipe consegue testar
 - Não testar mais código do que a equipe consegue implantar


Scrum e Kanban




- São Lean e ágeis
- Limitam o WIP
- Focam em fazer releases frequentes e rápidas
- Se baseiam em times “auto-organizados”
- Exigem a divisão das atividades em pequenas partes
- Sugerem a evolução contínua do processo de desenvolvimento

Kanban – trabalho prático

1. Defina seu fluxo de trabalho da sua equipe;
2. Divida o trabalho a ser feito em partes (itens), escreva cada item em um cartão e coloque em um quadro;
3. O nome das colunas indicam onde cada item está no fluxo de trabalho;
4. Limite a quantidade de trabalho em andamento para cada etapa do fluxo de trabalho.

Kanban – Exemplo real



JIRA
Dashboards ▾ Projects ▾ Issues ▾ Agile ▾ **Create**

Search





PWDA
Plan Work Report Board ▾


SPRINT: PWDA: Sprint 16 ▾
QUICK FILTERS: Story Bug Inbox Open Issues My Issues @none @cristiano @ghilardi @karina @lenz @utzig APP NUM MON


To Do Waiting For In Progress In Review Done


>  PWDA-1784 4 sub-tasks [E-mail] 17.Mar.15: Integrate Non-Drilling Simulator on APP


▾ Other Issues 55 issues


 **PWDA-2386**
 ↑ Support on WellPlot refactoring
 PWDA: New Wellplot [New GUI]


 **PWDA-2448**
 ↑ Tracer bullet to PWDA WellPlot
 PWDA: New Wellplot [New GUI]


 **PWDA-2302**
 ↑ Updates to real cases testing
 PWDA: New Flowcharts


 **PWDA-2446**
 ↑ Adjustments on the Tank Volume's checker on the Kick and Circulation Loss' flowcharts
 None


 **PWDA-2325**
 ↑ Convert BaseRealCaseTest to pytest
 None


 **PWDA-2445**
 ↑ Excessive use of memory
 None


 **PWDA-2379**
 ↑ Error combining "filter", "search" and editing Projects' Permission inside Projects' Tree


 **PWDA-1770**
 ↑ Implement the new Curve Collection Generator 2 using "true domain values" rather than
 None


 **PWDA-2371**
 ↑ The length of the curves error - 2.2.0 rc6
 None


 **PWDA-2289**
 ↑ Investigate why the mud weight didn't appear on the detailed graph of the 24h report
 None


 **PWDA-2224**
 ↑ Different data on the tracks for two viewers of the same project
 None


 **PWDA-2240**
 ↑ Investigate a situation where the calculations is not being performed
 None


 **PWDA-2054**
 ↑ Study and Implement another parameter for Calibration T&D
 PWDA: T&D Calibration

 **PWDA-2389**
 ↑ Tracer bullet to test javascript integration with Qt
 PWDA: New Wellplot [New GUI]

 **PWDA-2384**
 ↑ Update criteria for curve creation
 Kick without Circulation [Flowchart]

 **PWDA-2053**
 ↑ Add types of fluids (cathionic, SCOL, etc) in the rheological models
 None

 **PWDA-1824**
 ↑ Create a filter based on maximum DogLeg when collecting new inclination/azimut
 None

 **PWDA-2045**
 ↑ Insert NaN in the initialization of initial equipment pressure loss
 None

Kanban e Scrum - Vídeo