



# Kanban

Instituto Federal de São Paulo (IFSP)  
Campus Votuporanga



# Conteúdo da aula

- **Introdução**
- SCRUM
- Conclusão
- Exercícios propostos
- Referências bibliográficas



# Conteúdo da aula

- Introdução
- **SCRUM**
- Conclusão
- Exercícios propostos
- Referências bibliográficas



# Conteúdo da aula

- Introdução
- SCRUM
- **Conclusão**
- **Exercícios propostos**
- **Referências bibliográficas**

# Introdução

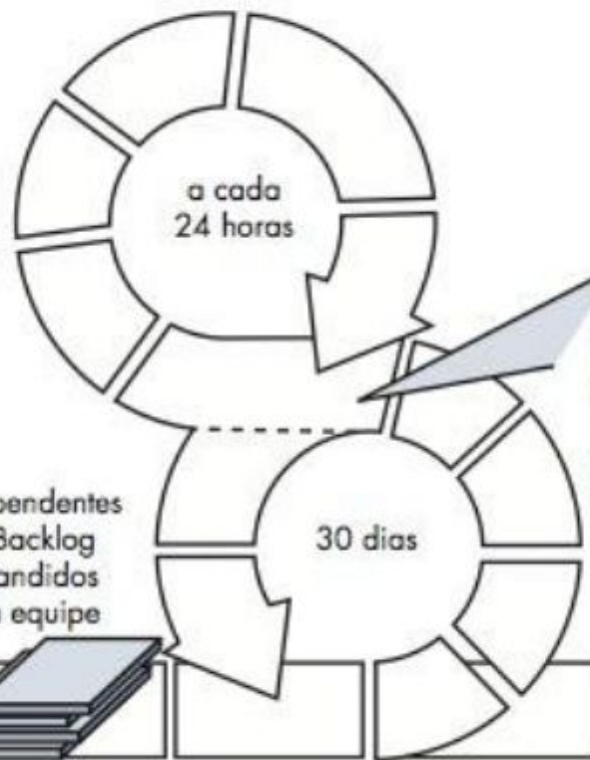


- Aulas Passadas ...
  - Características dos modelos de processo ágeis
    - Entregas constantes
    - Alta interação com o cliente
    - Equipes ágeis
    - Documentação necessária apenas

# Introdução

Aula passada ...

Modelo SCRUM



Scrum: Reuniões diárias de 15 minutos. Os membros da equipe respondem às questões básicas

- 1) O que você realizou desde a última Scrum?
- 2) Você está tendo alguma dificuldade?
- 3) O que você irá fazer antes da próxima reunião?

*Backlog do Sprint:*  
Funcionalidade(s)  
atribuída(s)  
ao sprint

Itens pendentes  
do Backlog  
expandidos  
pela equipe

**A nova funcionalidade  
é demonstrada no  
final do sprint**

*Backlog do Produto:*  
Priorização das funcionalidades do produto desejadas pelo cliente

# Introdução

- Apresenta visualmente o estado atual do processo de desenvolvimento



# Introdução



- Conceitos sobre o método Kanban
  -





# Conteúdo da aula

- Introdução
- **SCRUM**
- Conclusão
- Exercícios propostos
- Referências bibliográficas

# Kanban



- Breve Histórico do método Kanban
  - 1) Criação da ferramenta kanban pela Toyota para otimizar o processo
  - 2) Livro “Lean Software Development”
  - 3) Criação do método Kanban
  - 4) Livro ScrumBan

# Kanban



- **Aplicável a processos iterativos e incrementais**
  - Resulta em incrementos
  - Entrega contínua
- Objetivo de otimizar o processo de criação do software e melhorar a estrutura organizacional
  - Bom para otimização de curto prazo
  - Modificações “gigantes” planejadas a médio ou longo prazo

# Kanban



- **Requisitos para adotar o Kanban**

- Instalações físicas
- Volume de trabalho
- Modelo Colaborativo
- Motivação
- Foco em times



Explicação de cada requisito

# Kanban

- Requisitos para adotar o Kanban

- **Instalações físicas** ➡
- Volume de trabalho
- Modelo Colaborativo
- Motivação
- Foco em times

	A FAZER	FAZENDO	FEITO
PROJETO X	MOVIE MENU TELA BUSCA TOSH EMAIL SUI TE	TELA SINGLE TAG CLOUD BOTONE SHARE	HEADER TELA INDEX TELA PAGE LIST BOX TELA SEARCH TELA ASK
PROJETO Y	SCENE NAME ALERT SCENE NAME INFO	OPONS COS BOTÃO SHARE	INTRO MENU DUAL BOARD

Fonte: <https://br.pinterest.com/pin/388435536588901829/>

# Kanban



- **Requisitos para adotar o Kanban**

- Instalações físicas
- **Volume de trabalho** ➡
- Modelo Colaborativo
- Motivação
- Foco em times

## Explicação de cada requisito

Quantidade de tarefas relativamente grande

Mudança rápida do estado da tarefa (a fazer, finalizado)

# Kanban



- **Requisitos para adotar o Kanban**

- Instalações físicas
- Volume de trabalho
- **Modelo Colaborativo** →
- Motivação
- Foco em times

**Explicação de cada requisito**

Trabalho em equipes (ágeis) é  
essencial

# Kanban



- **Requisitos para adotar o Kanban**

- Instalações físicas
- Volume de trabalho
- Modelo Colaborativo
- **Motivação** ➡
- Foco em times

## Explicação de cada requisito

Equipe desejar trabalhar com a metodologia



# Kanban



- **Requisitos para adotar o Kanban**

- Instalações físicas
- Volume de trabalho
- Modelo Colaborativo
- Motivação
- **Foco em times** ➡

## Explicação de cada requisito

Autonomia para as equipes

Menos intervenção da chefia no trabalho das equipes

# Kanban



- Processo minimamente prescritivo, no qual são exigidos a menor quantidade pré-estabelecida de elementos possível
  - Papéis
  - Artefatos
  - Eventos
  - Atividades
- Inicialmente, usado para otimizar um processo já existente

# Kanban

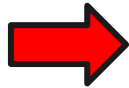


- **Fase da implementação do Kanban**

Tornando o trabalho visível
Limitando o trabalho em progresso
Lidando com retrabalho
Deixando o trabalho fluir
Reuniões frequentes
Cadência

# Kanban

- Fase da implementação do Kanban



## **Tornando o trabalho visível**

Limitando o trabalho em progresso

Lidando com retrabalho

Deixando o trabalho fluir

Reuniões frequentes

Cadência

# Kanban



- **Tornando o trabalho visível**
  - Criar um mapa visual do trabalho realizado no cotidiano
  - Dividido em três etapas
    - 1) Entender a natureza da demanda
    - 2) Entender o fluxo de valor
    - 3) Projetar o entendimento em um quadro

# Kanban

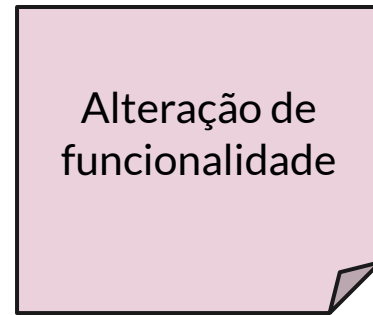
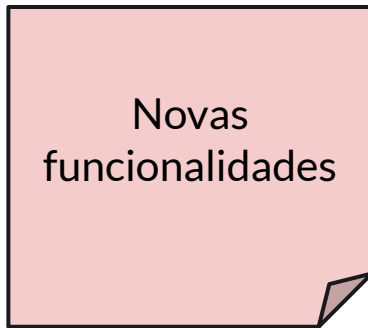


- **Tornando o trabalho visível**
  - 1) Entender a natureza da demanda
    - Determinar os diferentes tipos de trabalho da equipe
    - Cada tipo de trabalho é representado de maneira diferente
    - Os tipos de trabalho podem ser priorizados

# Kanban

- Tornando o trabalho visível

**Exemplo:** Equipe de manutenção

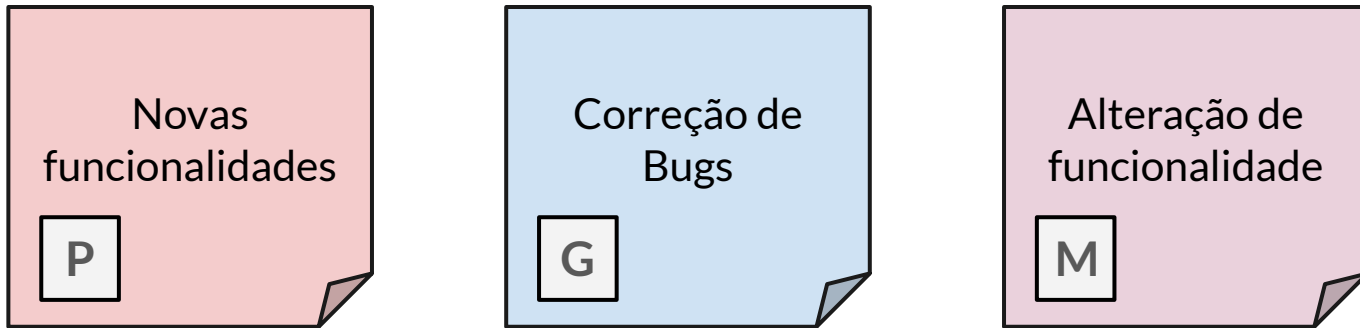


**Post-its de cores diferentes para representar cada tipo de demanda**

# Kanban

- Tornando o trabalho visível

**Exemplo:** Equipe de manutenção



**Barras verticais indicando priorização  
T-shirt sizing (Pequena, Média, Grande)**



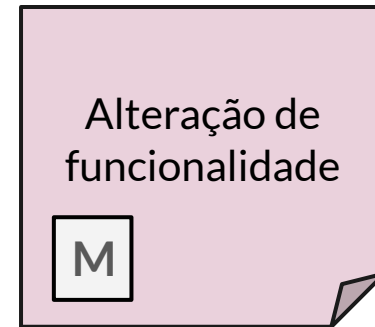
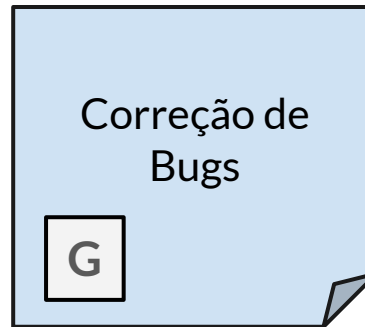
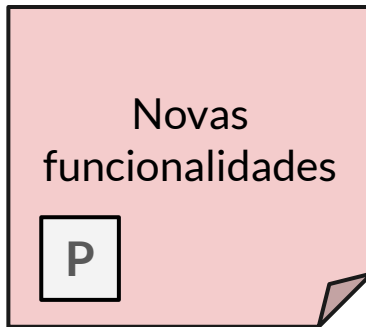
# Kanban

- Tornando o trabalho visível

## Outras formas de priorização:

- Origem da demanda
- Tipo de cliente

**Exemplo:** Equipe de manutenção



# Kanban

- Tornando o trabalho visível

- 2) Entender o fluxo de valor

- O fluxo de trabalho é do tipo **end-to-end**



# Kanban



- **Tornando o trabalho visível**
  - 2) Entender o fluxo de valor
    - Fugir do modelo tradicional de trabalho
      - **Modelo tradicional:** baseado em processos rígidos e pessoas com visões específicas do sistema
      - **Modelo Kanban:** baseado em apresentar a visão do todo e proporcionar entendimento global do sistema

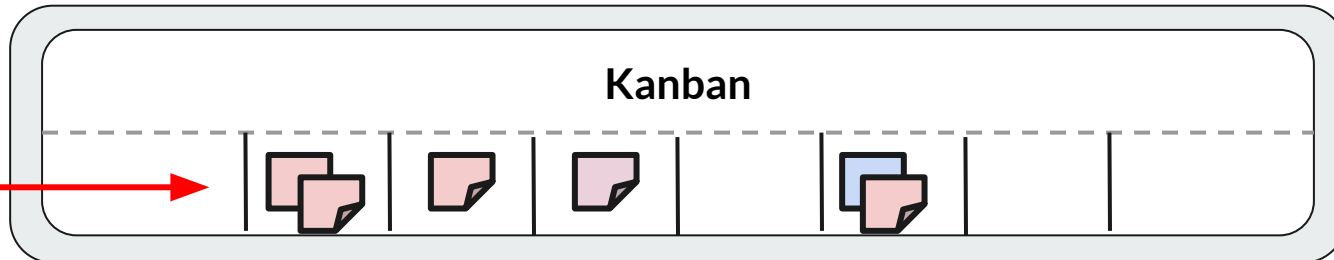
# Kanban

- Tornando o trabalho visível

- 2) Entender o fluxo de valor

- Para entender o fluxo de valor é necessário pensar em todos os **estados** que uma demanda pode passar

Estados



# Kanban



Estado: Descrição do estado

- **Tornando o trabalho visível**

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

F

G

H

I

J

K

# Kanban



- **Tornando o trabalho visível**

**Estado A:** Elaboração das Releases e funcionalidades

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

F

G

H

I

J

K

# Kanban



**Estado B:** Planejamento das Releases

- **Tornando o trabalho visível**

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

**B**

C

D

E

F

G

H

I

J

K

# Kanban



**Estado C:** Desenvolvimento

- **Tornando o trabalho visível**

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

F

G

H

I

J

K



# Kanban



**Estado D:** Subproduto desenvolvido

- **Tornando o trabalho visível**

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

F

G

H

I

J

K

# Kanban



**Estado E:** Inspeção por outro Dev.

- **Tornando o trabalho visível**

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

F

G

H

I

J

K

# Kanban



- **Tornando o trabalho visível**

**Estado F:** Inspecionado

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

**F**

G

H

I

J

K

# Kanban



**Estado G:** Avaliação pelo dono do produto

- **Tornando o trabalho visível**

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

F

G

H

I

J

K

# Kanban



- **Tornando o trabalho visível**

**Estado H:** Avaliado pelo dono do produto

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

F

G

H

I

J

K

# Kanban



- **Tornando o trabalho visível**

**Estado I:** Em homologação da Release

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

F

G

H

I

J

K

# Kanban



**Estado J:** Preparado para produção

- **Tornando o trabalho visível**

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

F

G

H

I

J

K

# Kanban



- **Tornando o trabalho visível**

**Estado K:** Em produção

- Exemplo: Estados no desenvolvimento de um software genérico

## Fluxo de trabalho baseado em cascata

A

B

C

D

E

F

G

H

I

J

**K**



# Kanban

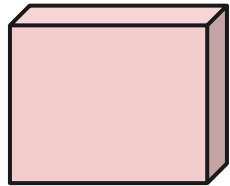


- **Tornando o trabalho visível**
  - Exemplo: Estados no desenvolvimento de um software genérico
    - O exemplo é baseado no modelo em cascata
    - O objetivo é implementar o modelo kanban e, gradativamente, mudar a cultura de trabalho

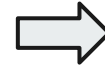
# Kanban

- **Tornando o trabalho visível**

- Exemplo: Estados no desenvolvimento de um software genérico



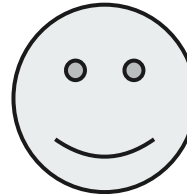
Release



Atividades



Funcionalidade



Avatar Dev.

# Kanban

- **Tornando o trabalho visível**



- Exemplo: Estados no desenvolvimento de um software genérico

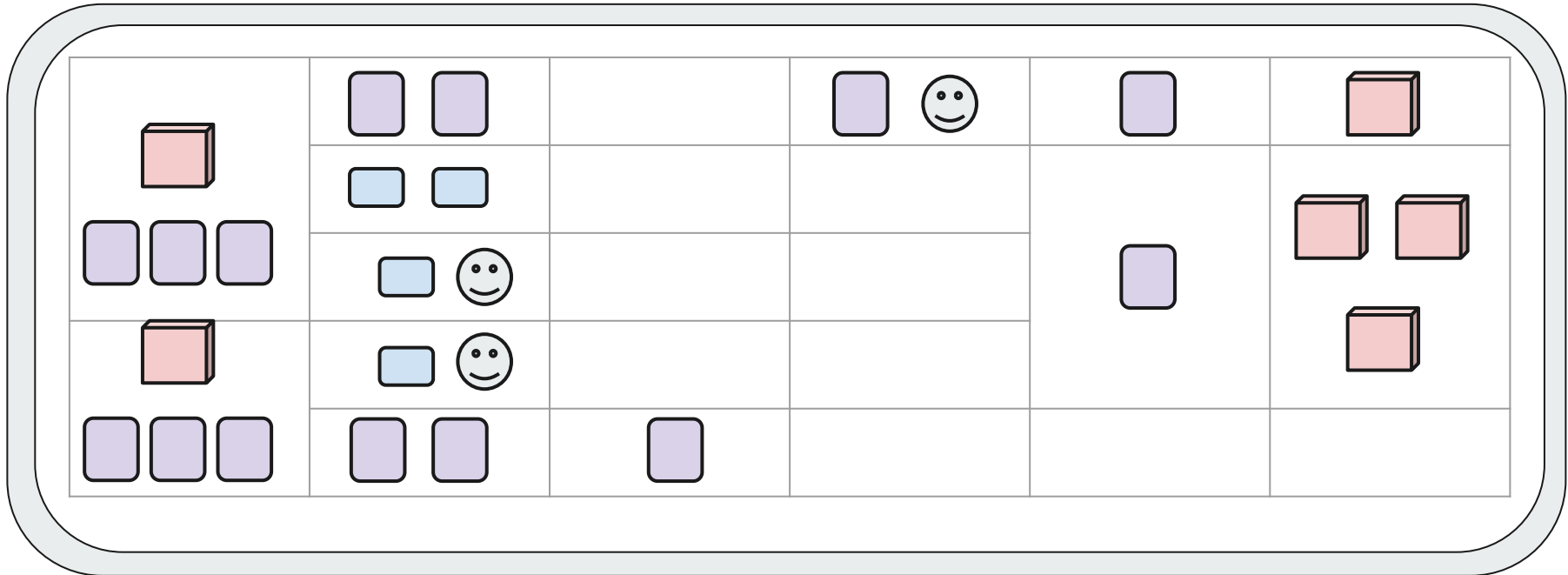
	Em Desen.	Em inspeção	Em avaliação	Em homologação	Preparado produção
Release Atual	Planejando				Em produção
	Em anda.				
	Feito				
Próximo Release	Desenvolvido	Inspecionado	Avaliado		

# Kanban

- **Tornando o trabalho visível**



- Exemplo: Estados no desenvolvimento de um software genérico



# Kanban



- **Tornando o trabalho visível**
  - 2) Entender o fluxo de valor
    - O mapa deve refletir fielmente a realidade
      - Evitar influência de padrões
      - Fugir do modelo ideal

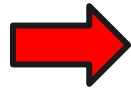
# Kanban



- **Tornando o trabalho visível**
  - 3) Projetar o entendimento do quadro
    - Criar o quadro visual

# Kanban

- Fase da implementação do Kanban



Tornando o trabalho visível
<b>Limitando o trabalho em progresso</b>
Lidando com retrabalho
Deixando o trabalho fluir
Reuniões frequentes
Cadência

# Kanban



- Limitando o trabalho em progresso
  - *Work In Progress (WIT)*
    - Definir limite de trabalho em cada estado do Kanban
    - Um número máximo de itens em cada estado
      - Não podem ser feitas transições de estados se o limite for atingido



# Kanban



- Limitando o trabalho em progresso
  - **Work In Progress (WIT)**
    - A limitação é embasada na **Teoria das Restrições (TOC)**

**Definição:** O sistema é limitado por seus componentes mais lentos. Não adianta ter uma produtividade alta em um componente se o seu resultado é processado por um componente com produtividade baixa.

# Kanban



- Limitando o trabalho em progresso
  - **Work In Progress (WIT)**
    - Está de acordo com o conceito de **Just in time (JIT)**
      - Apenas serão usados os recursos necessários e quando necessários
      - Evitar acúmulo de estoque

# Kanban



**OBS:** Os limites devem ser explicitados no quadro Kanban

- **Limitando o trabalho em progresso**

- **Work In Progress (WIP)**

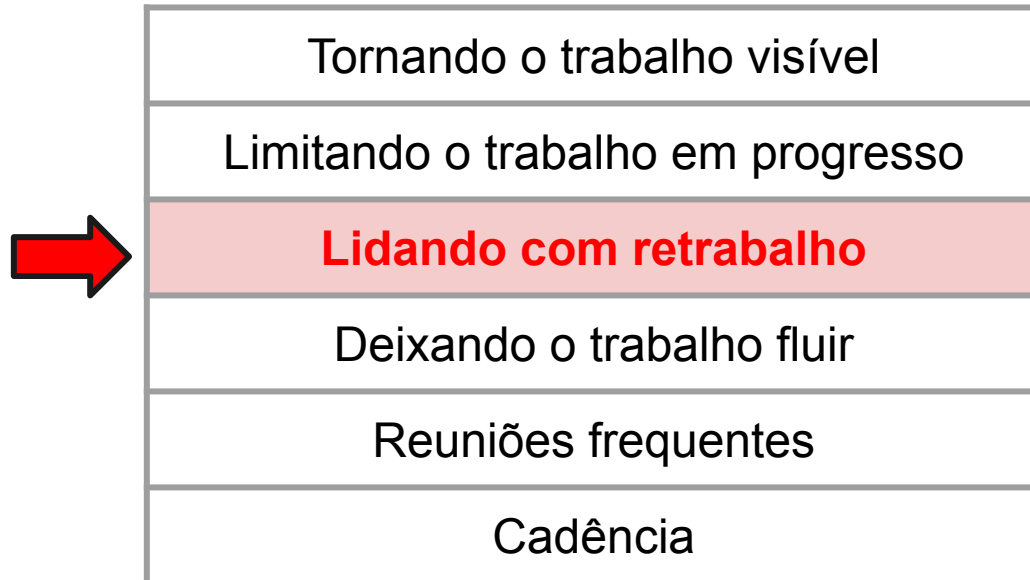
- **Benefícios:**

- “Lidar com os gargalos do sistema”
- Melhorar o fluxo de trabalho

- O limite é escolhido através de bom senso, experiência e análises empíricas

# Kanban

- Fase da implementação do Kanban



# Kanban



- **Lidando com retrabalho**

- Algum cartão (atividade ou funcionalidade) deve retornar a algum estado anterior

Como lidar com o retrabalho deste cartão?

# Kanban



- **Lidando com retrabalho**

- Algum cartão (atividade ou funcionalidade) deve retornar a algum estado anterior

Como lidar com o retrabalho deste cartão?

- ***Uma solução:*** Não mover o cartão e acionar a política denominada Stop the line (Parada de linha)

# Kanban



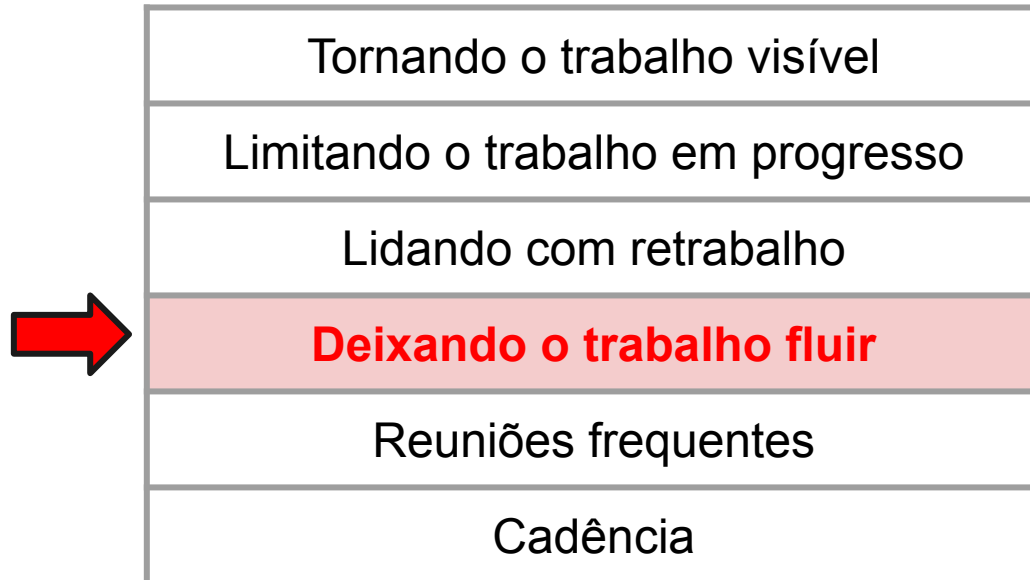
- **Lidando com retrabalho**

- **Stop the line**

- Apresenta um bloqueio no fluxo de trabalho
- Execução de tratamento imediato
- Uso de cartões diferentes no quadro Kanban indicando o bloqueio no fluxo de trabalho
- Preenchimento excessivo do quadro Kanban, gerando uma cultura de evitar o retrabalho

# Kanban

- Fase da implementação do Kanban





# Kanban



- **Deixar o trabalho fluir**
  - Modelo Kanban incentiva a criação de equipes ágeis
    - Auto-organizadas
    - Dispostas a resolver problemas
    - Autonomia na definição das tarefas
    - Colaborativas

# Kanban



- **Deixar o trabalho fluir**

Após transacionar um cartão, o que o desenvolvedor deve fazer?

# Kanban



- **Deixar o trabalho fluir**

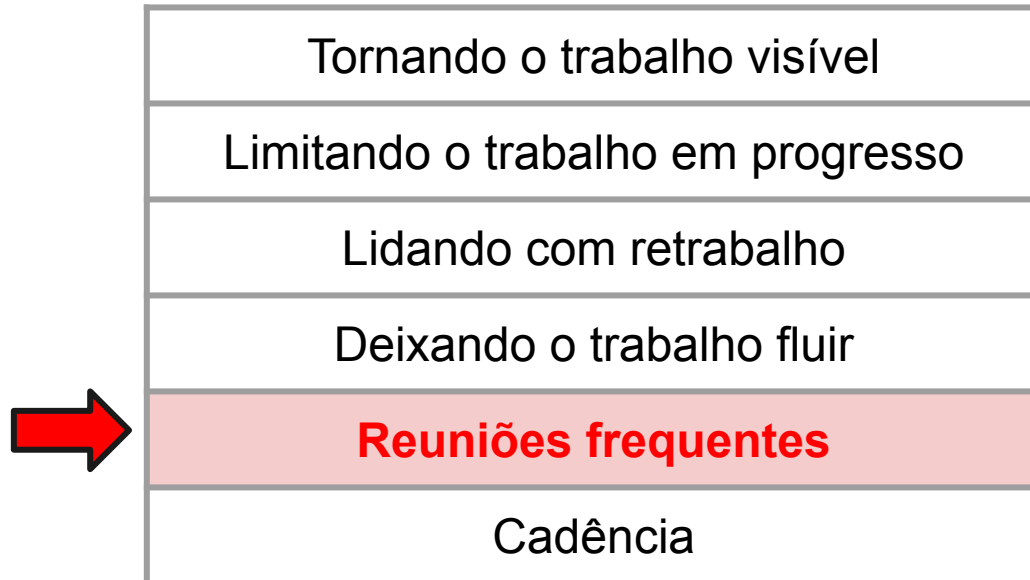
Após transacionar um cartão, o que o desenvolvedor deve fazer?

**Resposta**: Deve analisar o quadro e determina qual a atividade irá fazer o fluxo de trabalho se manter contínuo

**OBS**: Buscar e resolver gargalos no quadro

# Kanban

- Fase da implementação do Kanban



# Kanban

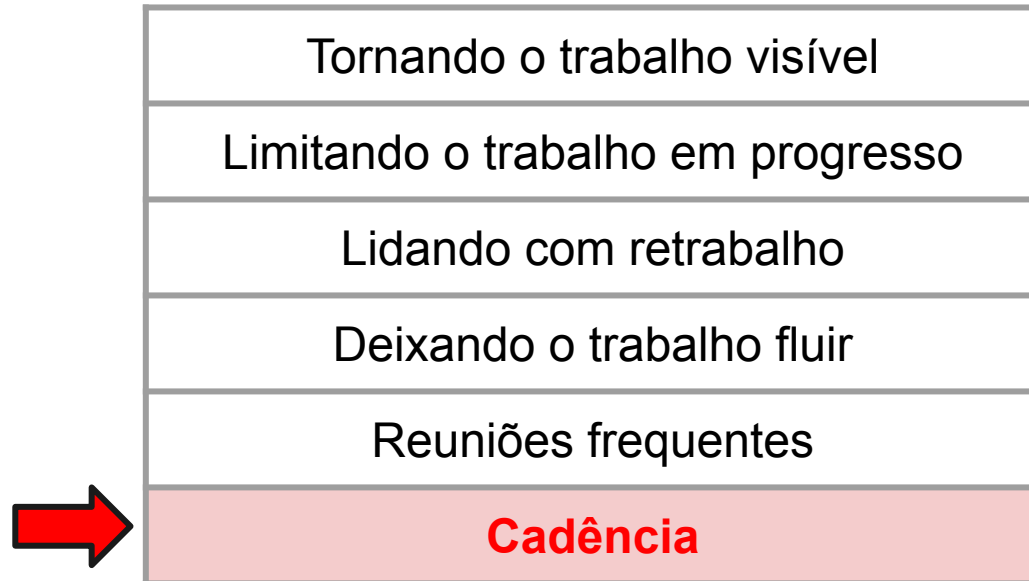


- **Reuniões frequentes em frente ao quadro**
  - Discussão do trabalho em frente ao quadro
  - Reuniões diárias e disseminação de conhecimento
  - Reuniões curtas (Semelhante ao SCRUM)
  - Decisão de qual atividade realizar

# Kanban



- **Fase da implementação do Kanban**



# Kanban



- **Cadência**

- **Definição**: ritmo em que os eventos ocorrem
- A cadência do Kanban pode ser decidida pela equipe
  - No SCRUM os tempos dos eventos são mais pré-determinados
  - No Kanban os tempos podem ser mais flexíveis para se adaptar ao desenvolvimento



# Conteúdo da aula

- Introdução
- SCRUM
- **Conclusão**
- **Exercícios propostos**
- **Referências bibliográficas**



# Conclusão



## Exercícios propostos



1) A atividade consiste em realizar um Mini Kanban. Escolha um sistema computacional em algum contexto. Determine as releases e funcionalidades que devem ser implementadas. Distribua elas em estados. Tais estados podem estar divididos em subestados. Garanta que a escolha dos estados forneça um fluxo end-to-end. Crie um quadro Kanban representando um “momento” durante a execução do programa.

## Referências bibliográficas Básicas



- CRUZ, Fabio. Scrum e Agile em Projetos (2a. edição): guia completo. 2. ed. Rio de Janeiro:Brasport. 2018.
- PRIKLADNICKI, Rafael; WILLI, Renato, MILANI, Fabiano. Métodos Ágeis para Desenvolvimento de Software. Porto Alegre:Bookman. 2014.
- BELL , P.; BEER , B. Introdução ao GitHub: um Guia que Não é Técnico. 1. ed. São Paulo: Editora Novatec. 2014.

## Referências bibliográficas Complementares



- BROD, Cesar. SCRUM: Guia prático para projetos ágeis. 2ª. ed. São Paulo:Novatec. 2015.
- KERZNER, H. Gerenciamento de Projetos: uma abordagem sistêmica para planejamento, programação e controle. 10ª. ed. São Paulo:Blucher. 2011.
- OLIVEIRA, Bruno Souza de. Métodos Ágeis e Gestão de Serviços de TI. Rio de Janeiro:Brasport. 2018.

## Referências bibliográficas Complementares



- SCRUMSTUDY. Um Guia para o Conhecimento em Scrum (Guia SBOK™). ed. 2016. Phoenix. 2016. Disponível em <http://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016-Portuguese.pdf>.
- PROJECT MANAGEMENT INSTITUTE (PMI). Um guia do conhecimento em gerenciamento de projetos. 5. ed. PMI:Newton Square. 2013.

# Material de Apoio



- PRESSMAN, R.S. Engenharia de Software: Uma Abordagem Profissional. 7. ed. São Paulo: McGraw-Hill, 2011.
- Site contendo o manifesto ágil. Acessado em 06/03. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>

# Imagens



- <https://br.pinterest.com/pin/388435536588901829/>