



# Bem vindo(a)s!!!



**Lanche seco, nunca mais!**



***Link da Turma***



# Tópicos Essenciais à Programação

- Análise e Desenvolvimento de Sistemas - **UNIPÊ / Estácio SA**
- MBA Engenharia e Administração de Dados - **UNIESP**
- Msc. Tecnologia da Informação - **IFPB**(Tendendo não desistir)
  - Pesquisador Projeto ANATEL
- Engenheiro de Dados Sr. - **NTT Data**
- Professor Membro - Artificial Intelligence & Data Analytics - **AIDA**
- Professor Membro - **GDG João Pessoa**
- Professor - **UNIESP**
- Criador de Conteúdo - **Youtube | Nerd Data Lab**



<https://beacons.ai/wuldsom>



# T.E.P – O que vamos aprender?

- 1        \* Lab - Apresentação da disciplina / Introdução aos comandos linux/Windows
- 2        \* Lab - comandos linux/Windows + Introdução ao Github
- 3
- 4        \* Lab - Github Configuração do ambiente, criação de conta
- 5        \* Lab - Github - Versionamento e acessos externos
- 6        \* Lab - Github - Dúvidas + Práticas
- 7
- 8        \* Sala - Básico DevOps (teoria)
- 9        \* Palestra “Quais Soft Skills devo melhorar para seguir na área de TI?”
- 10      \* Sala - Metodologias Ágeis (Scrum e Kanban) I e II
- 11      \* Sala - Básico Teste de Software
- 12      \* Sala - SEO - Search Engine Optimization
- 13      }
- 14



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

- \* Exercícios
- \* Avaliações Escritas + Projetos

}



1

2

3

4

5

6

7

8

9

10

11

12

13

14

**01** Metodologias  
História;  
Cascata / Ágeis;

**02** Scrum

**03** Kanban



{}

# Gerenciamento de Projetos

- ▶ Orientado a processos
  - ▶ Processos bem definidos devem ser impostos para garantir a qualidade do produto
- < ▶ Rígido
  - ▶ Pressupõe que é possível especificar de antemão todos os requisitos do projeto
- ▶ Preditivo
  - ▶ Cada etapa de desenvolvimento é baseada na etapa anterior, parte do princípio de que requisitos são estáveis
- ▶ Burocrático
  - ▶ Sobrecarregar desenvolvimento, pode comprometer a velocidade do projeto
- ▶ Possui forte resistência a mudanças

{}



# Gerenciamento de Projetos de Software

## ► Particularidades

&lt;

- Invisibilidade
  - Progresso não é imediatamente visível
- Complexidade
- Flexibilidade
  - Propenso a um alto grau de mudança
- Dificuldade de antever suas funcionalidades
- Necessidades surgem durante seu desenvolvimento, e vão amadurecendo até a sua implantação

}

- A **mudança** se torna inevitável



1

2

3

4

5

6

3

1

—

—

12

15

01 {

# [História]

- “[É] uma palavra derivada de “**método**”, do Latim “methodus” cujo significado é “caminho ou a via para a realização de algo”;
  - Existem vários tipos: ensino, científica, pesquisa, ágil...
  - “Método é o processo para se atingir um determinado fim ou para se chegar ao conhecimento;

>



**uniesp**  
Centro Universitário

1

2

3

4

5

O conceito de *metodologia ágil* que conhecemos hoje surgiu em 2001. Como reação às abordagens de gerenciamento de projetos em cascata (aqueles que organizam o desenvolvimento de aplicações como uma série de sequências lineares), um grupo de desenvolvedores redigiu o Manifesto for Agile Software Development.

<http://agilemanifesto.org/iso/ptbr/manifesto.html>

{

# 01 { 2 História 3 Metodologia Ágil]



13

14

1

01 {

2

3

# Declaração de Interdependência

4

5

6

- ▶ Abordagens ágeis e adaptativas para permitir ligar pessoas, projetos e valor

7

8

9

10

“Somos uma comunidade de **líderes de projeto** que é altamente eficaz entregando resultados.”

11

12

13

}

14



1      01 {  
2

3      0 que significa  
4      interdependência?  
5

- 6
- 7      ▶ Que membros de uma equipe de projeto são parte **interdependente**  
8      do todo e não um grupo de **indivíduos desconectados**.
    - 9            ▶ Dependem reciprocamente uns dos outros
  - 10     ▶ Que equipes de projeto, seus clientes, seus interessados  
11      (*stakeholders*) também são interdependentes.
- 12

{  
13

}

}  
14

1

2

3

4

5

6

7

8

9

10

11

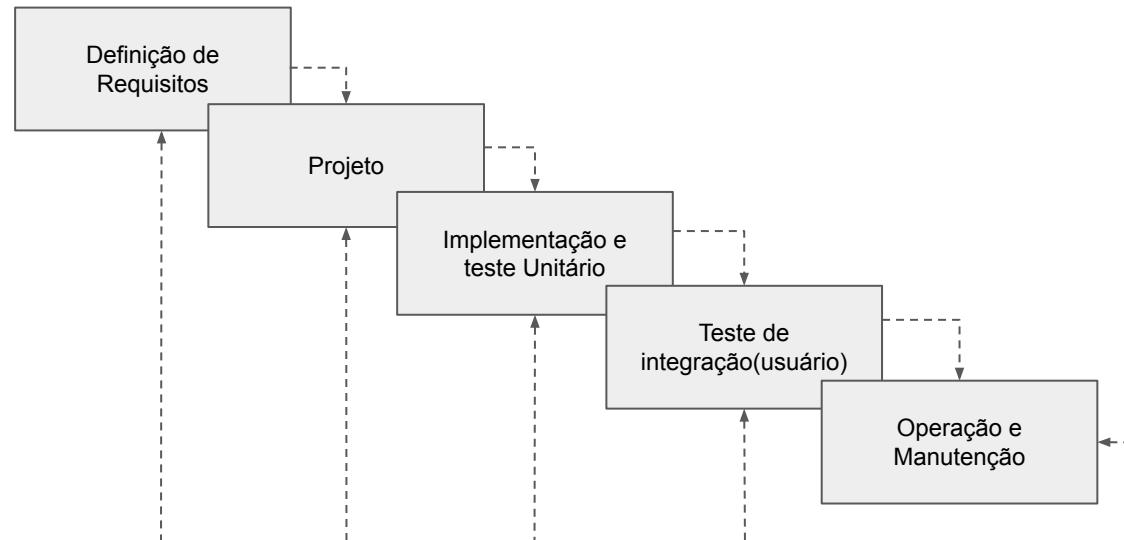
12

13

14

# 01 {

## [Metodologia Cascata]



}

1

2

3

4

5

6

7

8

9

10

11

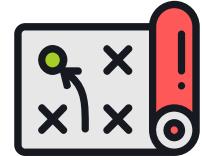
12

13

14

# 01 {

## [Metodologia Ágil]

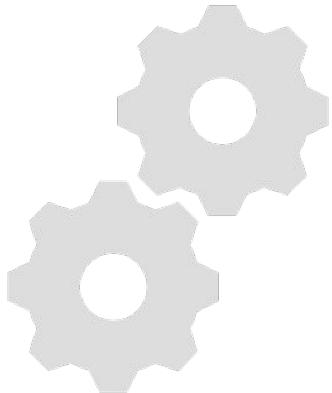


Conjunto de práticas que visam permitir, por meio de inspeção e adaptações frequentes, entregas rápidas, com qualidade e alinhadas à necessidade do cliente e da empresa. Além disso, tem como valor básico os indivíduos e suas interações, mais do que processos e ferramentas



Os métodos tradicionais de desenvolvimento de software são lentos e atrasam as entregas de produtos. Em um mundo de TI dinâmico, pequenos atrasos podem causar repercussões negativas e outros prejuízos a nível financeiro.





# Gerenciamento Ágil de Projetos de Software

# Gerenciamento Ágil de Projetos

- ▶ Um conjunto de **valores, princípios e práticas** que **auxiliam** a equipe de projeto a entregar produtos ou serviços de valor em um ambiente **complexo, instável e desafiador**;
- ▶ É o exercício de princípios e práticas ágeis aliados aos conhecimentos, habilidades e técnicas na elaboração das atividades de projeto, de forma a diminuir o **time-to-market**, e se adequar às mudanças durante o projeto.

# Gerenciamento Ágil de Projetos

- ▶ Objetivo
- ▶ Garantir que exista um equilíbrio entre demandas de qualidade, escopo, tempo e custos

# Gerenciamento Ágil de Projetos

- ▶ Valores centrais
  - ▶ As **respostas às mudanças** são mais importantes que o segmento de um plano
  - ▶ A **entrega de produtos** está acima da entrega de documentação
  - ▶ **Priorização da colaboração** do cliente sobre a negociação de contratos
  - ▶ Os **indivíduos e suas interações** são mais importantes que os processos e ferramentas

# Gerenciamento Ágil de Projetos

- ▶ Principais objetivos
- ▶ **Inovação contínua:** a idéia de inovação é associada a um ambiente cuja cultura estimule o auto-gerenciamento e a autodisciplina
- ▶ **Adaptabilidade do produto:** os produtos adaptáveis às novas necessidades do futuro

# Gerenciamento Ágil de Projetos

- ▶ Principais objetivos
- ▶ **Tempos de entrega reduzidos:** direcionamento preciso e capacidade técnica da equipe
- ▶ **Capacidade de adaptação do processo e das pessoas:** equipe confortável com mudanças, processo leve
- ▶ **Resultados confiáveis:** entrega de produtos que garantam operação, crescimento e lucratividade da empresa

# Técnicas de Gerenciamento Ágil de Projetos

- ▶ Foque nas pessoas
  - ▶ As pessoas e a maneira como interagem são determinantes mais importantes para o sucesso de um projeto
- ▶ Organize seu projeto em iterações
  - ▶ Curtos períodos de tempo onde ao seu final chega-se a um objetivo específico
- ▶ Estabeleça marco de entrega final somente se for realmente necessário\*

# Técnicas de Gerenciamento Ágil de Projetos

- ▶ Tenha um plano de projeto de alto nível
  - ▶ Principais dependências externas, iterações planejadas e uma estimativa de término (se possível)
- ▶ Crie planos de iteração detalhados com base no JIT (Just In Time)
  - ▶ Você só pode planejar precisamente com algumas semanas de antecedência da realização

# Técnicas de Gerenciamento Ágil de Projetos

- ▶ Faça estimativa de coisas pequenas
  - ▶ É mais fácil fazer a estimativa de um trabalho que levará apenas um dia do que estimar algo que levará um mês.
- ▶ As pessoas deveriam estimar seu próprio trabalho
  - ▶ As melhores estimativas vêm de baixo para cima e não de cima para baixo

# Gerenciamento Ágil de Projetos

- ▶ Críticas

- ▶ Dificuldade de manutenção pela falta de documentação \*
- ▶ Efetividade da programação em pares: custo x benefício
- ▶ Dificuldade de se ter o cliente no local
- ▶ Dificuldade de estabelecer contrato com escopo variável
- ▶ Requer colaboração e confiança entre equipe e cliente

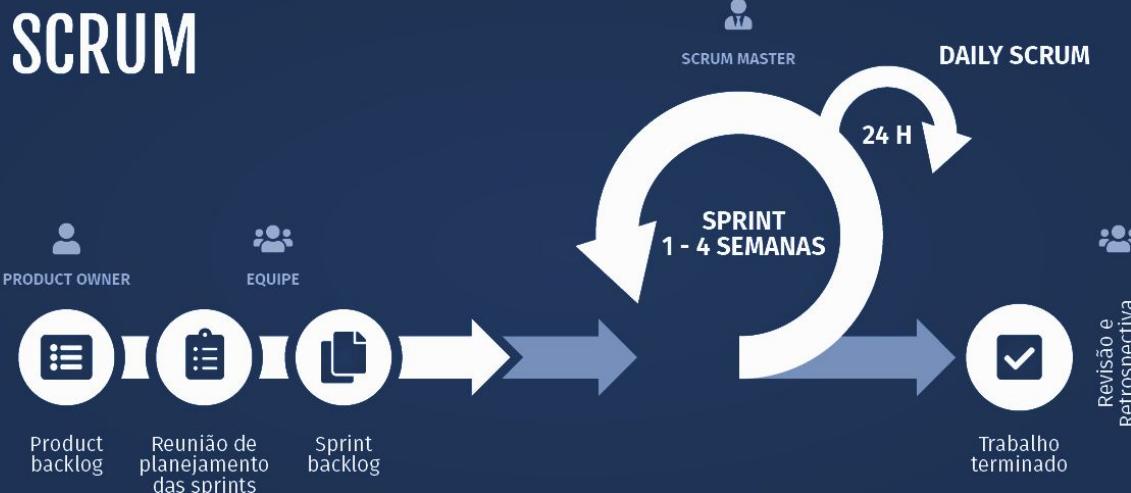
# Abordagem Clássica vs. Abordagem Ágil

	Clássica	Ágil
Desenvolvedor	hábil	Ágil (e hábil)
Cliente	pouco envolvido	comprometido
Requisitos	conhecidos, estáveis	emergentes, mutáveis
Retrabalho	caro	barato
Planejamento	direciona resultados	resultados o direcionam
Foco	grandes projetos	projetos de natureza exploratória e inovadores
Objetivo	controlar, em busca de alcançar o planejado	simplificar processo de desenvolvimento

# Abordagem Clássica vs. Abordagem Ágil

- ▶ Ciclo de vida ágil é semelhante ao clássico
  - ▶ Define o que o cliente quer e inicia o projeto
  - ▶ Planeja o projeto, calculando o esforço
  - ▶ Executa o plano, construindo a solução
  - ▶ Monitora resultados e entrega ao cliente

# 01 { [ SCRUM ]



1

2

3 • SCRUM

4

- 5     ○ O Scrum é um framework de gerenciamento que as equipes usam para se  
6       **auto-organizar** e trabalhar em direção a um objetivo em comum. A  
7       estrutura descreve um conjunto de **reuniões**, ferramentas e funções  
8       para uma entrega eficiente de projetos.
- 9
- 10    ○ Assim como um time esportivo treina para um jogo importante, as  
11      práticas do Scrum permitem que as equipes promovam o  
12      autogerenciamento, aprendam com base nas experiências e se adaptem  
13      às mudanças.
- 14    ○ As equipes de software usam o Scrum para solucionar problemas  
15      complexos de maneira econômica e sustentável.



1

2

3     ● **SCRUM - Princípios**

4

5         ○ Transparência:

6

- 7             ■ As equipes trabalham em um ambiente no qual todos têm  
8             consciência dos desafios que os outros podem estar enfrentando.
- 9

10         ○ Reflexão:

11

- 12             ■ Incorporam-se os pontos de reflexão frequentes ao framework para  
13             permitir que os membros da equipe revisem o próprio progresso.
- 14

15         ○ Adaptação:

16

- 17             ■ Os membros da equipe podem redefinir a prioridade das tarefas  
18             conforme as mudanças nos requisitos do cliente.
- 19



## 1 SCRUM - Valores

- **Compromisso:**

- Os membros da equipe Scrum se comprometem com tarefas e objetivos baseados em tempo e se dedicam a melhorar continuamente para encontrar a melhor solução.

- **Coragem:**

- As equipes Scrum demonstram coragem ao fazer perguntas abertas e desafiadoras. Elas têm discussões honestas e transparentes para chegar à melhor solução.

- **Foco:**

- O foco estará nas tarefas(product backlog) selecionadas para fornecer resultados dentro de um prazo limitado.

- **Foco:**

- O foco estará nas tarefas(product backlog) selecionadas para fornecer resultados dentro de um prazo limitado.

- **Abertura:**

- Os membros da equipe Scrum são receptivos a novas ideias e oportunidades que corroboram com o aprendizado individual e a qualidade geral do projeto.

- **Respeito:**

- Os membros da equipe respeitam os gerentes de projeto, os outros membros e o processo Scrum.



1

## SCRUM - Artefatos

- **Product backlog:**

- O product backlog é uma lista dinâmica de recursos, requisitos, melhorias e correções que devem ser concluídos para que projeto seja bem-sucedido. Em essência, é a lista de tarefas da equipe, que constantemente é revisitada e passa por redefinições de prioridade para se adaptar às mudanças do mercado.

6

- **Sprint backlog:**

- O Sprint backlog é a lista de itens a serem concluídos pela equipe de desenvolvimento no ciclo atual do Sprint. Antes de cada Sprint, a equipe escolhe em quais itens do product backlog trabalhará.

10

- **Incremento:**

- O incremento é um passo em direção a uma meta ou visão. É o produto final utilizável de um Sprint. As equipes podem adotar diferentes métodos para determinar e demonstrar suas metas de Sprint.

14



1

## 2 SCRUM - Papéis

### 3 • Responsável pelo produto - Product Owner - P.O:

- 4 ○ O responsável pelo produto se concentra em garantir que a equipe de desenvolvimento  
forneça o máximo de valor aos negócios.
  - 5 ■ Fornecem à equipe uma orientação clara sobre quais recursos entregaráo em  
seguida.
  - 6 ■ Fazem a ponte entre o que a empresa quer e o que a equipe entende.
  - 7 ■ Decidem quando e com que frequência os resultados serão apresentados.

### 9 • Líder Scrum - Agile Master - AM:

- 10 ○ Os líderes Scrum representam o Scrum dentro das equipes. São responsáveis pela  
eficácia da equipe Scrum. Os líderes Scrum também são responsáveis por:
  - 11 ■ Planejar os recursos necessários para cada Sprint.
  - 12 ■ Promover outros eventos de Sprint e reuniões de equipe.
  - 13 ■ Conduzir a transformação digital na equipe.
  - 14 ■ Promover treinamentos da equipe ao adotar novas tecnologias.
  - Comunicar-se com grupos externos para solucionar possíveis desafios que a equipe  
esteja enfrentando como um todo.



1

2

### 3 SCRUM - Papéis

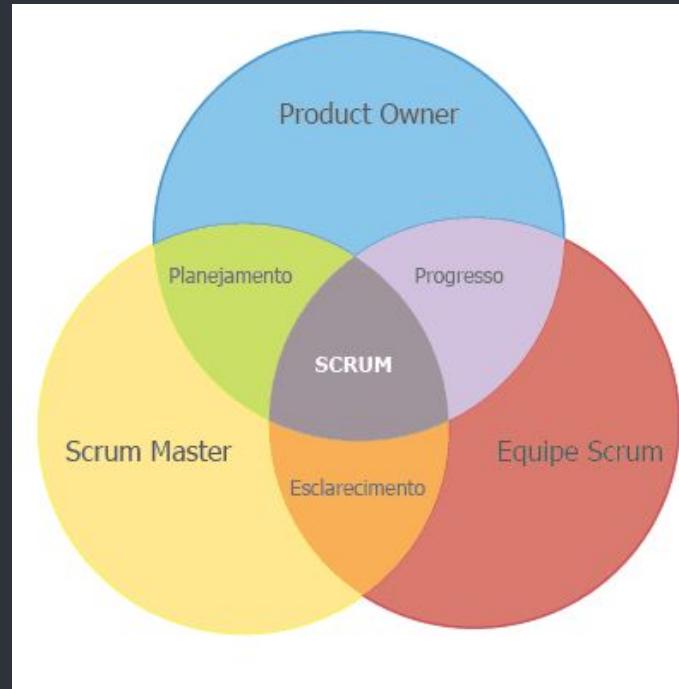
4

- 5     ● **Equipe de desenvolvimento Scrum:**

- 6         ○ O incremento é um passo em direção a uma meta ou visão. É o produto final utilizável  
7             de um Sprint. As equipes podem adotar diferentes métodos para determinar e demonstrar  
8             suas metas de Sprint.
  - 9                 ■ A equipe Scrum é composta por testadores, designers, especialistas em experiência  
10                 do usuário, engenheiros de operações e desenvolvedores.
  - 11                 ■ Trabalham de maneira colaborativa para garantir a conclusão eficaz do Sprint.
  - 12                 ■ Defendem práticas de desenvolvimento sustentável.
  - 13                 ■ São auto-organizadas e efetuam os projetos com um senso de coletividade evidente.
  - 14                 ■ Orientam o planejamento e a estimativa de quanto trabalho conseguem concluir em  
                   cada Sprint.

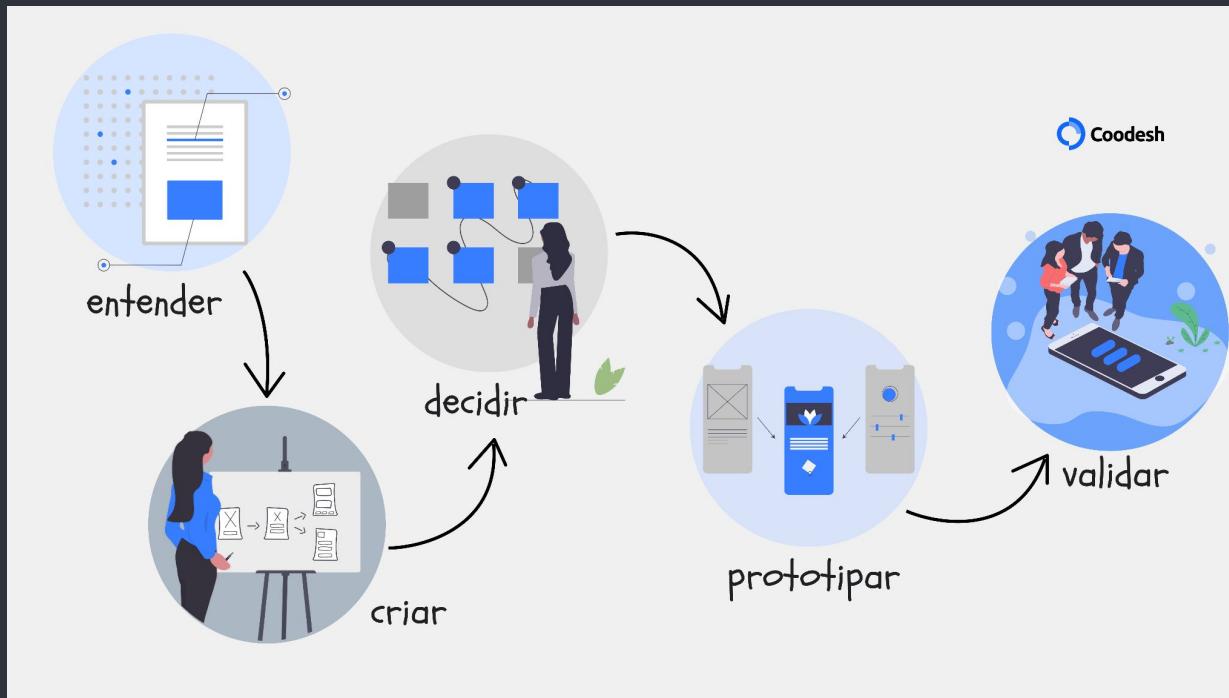


1  
2  
3  
4  
5  
6 }  
7  
8  
9  
10  
11  
12  
13  
14 }



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

# [CERIMÔNIAS]



1  
2

### 3 SCRUM - Cerimônias = PLANEJAMENTO

4  
5  
6  
7  
8

- Relativamente curto
- Projeto da arquitetura do sistema
- Estimativas de datas e custos
- Criação do backlog
  - Participação de clientes e outros departamentos
  - Levantamento dos requisitos e atribuição de prioridades
- Definição de equipes e seus líderes
- Definição de pacotes a serem desenvolvidos



1

2

## 3 SCRUM - SPRINT

4

- O time recebe uma parte do backlog para desenvolvimento
- O backlog não sofrerá modificações durante o Sprint
- Duração de 1 a 4 semanas
- Sempre apresentam um executável ao final

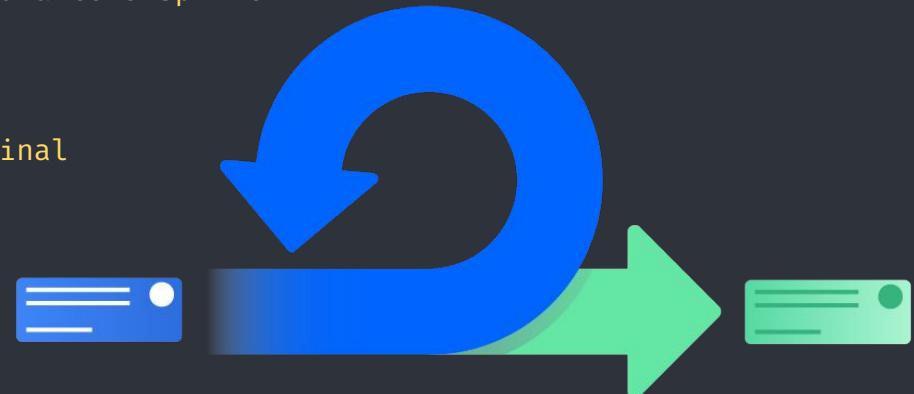
10

11

12

13

14



1

## 2 SCRUM - DAILY

3

- Cerca de 15 minutos de duração
- Todos respondem às perguntas:
  - O que você realizou desde a última reunião?
  - Quais problemas você enfrentou?
  - Em que você trabalhará até a próxima reunião?
- Benefícios:
  - Maior integração entre os membros da equipe
  - Rápida solução de problemas
    - Promovem o compartilhamento de conhecimento
  - Progresso medido continuamente
    - Minimização de riscos



## 1 SCRUM - Review



Deve obedecer à data de entrega

Permitida a diminuição de funcionalidades



Apresentação do produto ao cliente

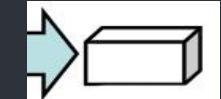
Sugestões de mudanças são incorporadas ao backlog



Benefícios:

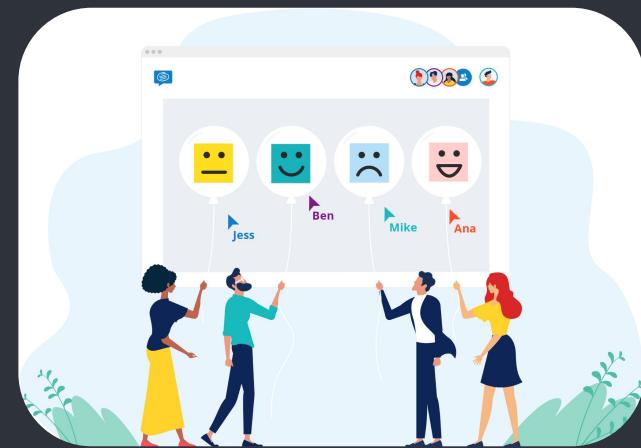
- Apresentar resultados concretos ao cliente
- Integrar e testar uma boa parte do software
- Motivação da equipe

Nova Funcionalidade



1    **SCRUM - Retro - Retrospectiva do Sprint**

- 2
- 3    ▶ Não deve ser maior do que 3 horas
- 4
- 5    ▶ Participam desta reunião  
      ▶ Time, ScrumMaster e, opcionalmente, Product Owner
- 6
- 7    ▶ Os membros do time devem responder a duas questões:  
      ▶ O que aconteceu de bom durante o último Sprint?  
      ▶ O que pode ser melhorado para o próximo Sprint?
- 8
- 9    ▶ ScrumMaster escreve as respostas e prioriza na ordem  
      que deseja discutir as potenciais melhorias
- 10
- 11    ▶ ScrumMaster nesta reunião tem o papel de fazer com que  
      o time encontre melhores formas de aplicar o Scrum
- 12
- 13
- 14



1    **SCRUM - Encerramento**

2

3

4

5    ▶ **Finalização do projeto**

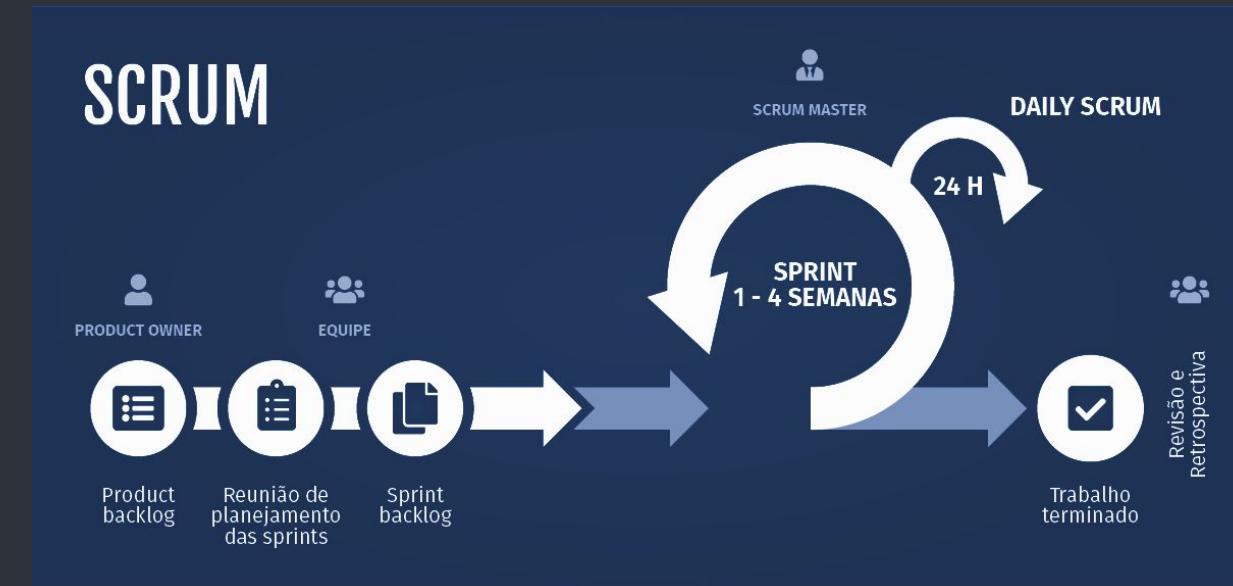
6    ▶ Atividades:

- 7
- 8       ▶ Testes de integração
- 9       ▶ Testes de sistema
- 10      ▶ Documentação do usuário(técnica ou de negócio)
- 11      ▶ Preparação de material de treinamento
- 12      ▶ Preparação de material de marketing
- 13
- 14



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

}



# 03 { [KanBan]

< Uma estratégia para otimizar o fluxo de valor para os **stakeholders** através de um processo que utiliza facilitação visual e limitação de work-in-progress (trabalho em progresso) de um sistema puxado.

&gt;



1

2

3

4

## [História]

5

- Originalmente, o kanban (com k minúsculo) foi introduzido por Taiichi Ohno na indústria de manufatura japonesa em 1940, mais especificamente dentro do Sistema Toyota de Produção.
- No Brasil, ele está em uso há várias décadas, não apenas na indústria, como em qualquer empreendimento que deseje alcançar um nível de excelência em suas atividades.



13

14

1

2

3

4

5

6

7

8

9

10

11

12

13

14

# [Princípios]

Princípios do Gerenciamento de Mudanças	Princípios da Entrega de Serviços
Comece com o que você faz agora	Foco nas necessidades e expectativas do cliente
Concorde em Buscar Mudanças Incrementais e Evolutivas	Gerencie o trabalho, não os trabalhadores
Incentivar atos de liderança em todos os níveis	Revise regularmente a rede de serviços



1

## 2 Princípios do Gerenciamento de Mudanças

- 3 • Comece com o que você faz agora:
  - 4 ○ Flexibilidade de usar o método em
  - 5 cima de fluxos de trabalho, sistemas
  - 6 e processos existentes sem
  - 7 interromper o que já está em vigor.
- 8 • Concordar em Buscar Mudanças Incrementais
- 9 e Evolutivas:
  - 10 ○ Ele incentiva pequenas mudanças
  - 11 incrementais e evolutivas contínuas
  - 12 no processo atual, implementando
  - 13 formulários de colaboração e
  - 14 feedback.
- 15 • Incentivar Atos de Liderança em Todos os
- 16 Níveis:
  - 17 ○ A liderança em todos os níveis
  - 18 deriva das percepções cotidianas das
  - 19 pessoas e atua para melhorar sua
  - 20 forma de trabalhar.

## Princípios da entrega de serviços

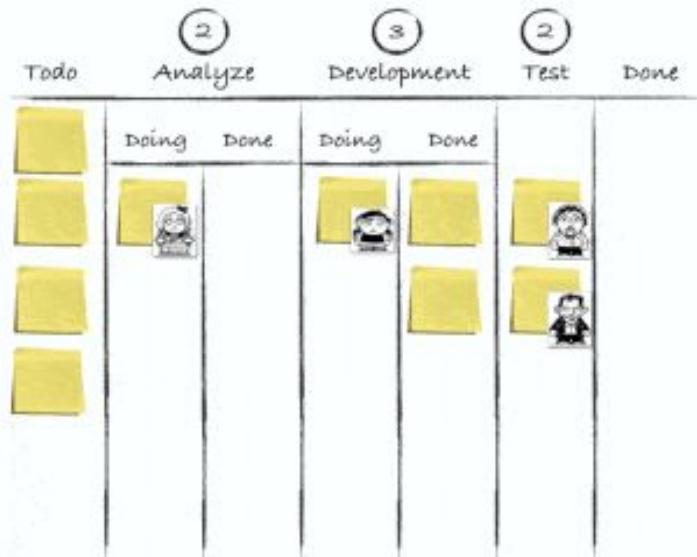
- Foco nas necessidades e expectativas do cliente:
  - Entregar valor ao cliente deve estar no centro de cada organização.
- Gerenciar o Trabalho:
  - Gerenciar o trabalho em sua rede de serviços garante que você capacite as habilidades das pessoas para se auto-organizarem em torno do trabalho.
- Rever regularmente a Rede de Serviços:
  - Uma vez desenvolvida, uma abordagem orientada a serviços requer avaliação contínua para promover uma cultura de atendimento ao cliente.



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

&lt;

## [Práticas]



From KANBAN IN ACTION by MARCUS HAMMARBERG and JOAKIM SUNDÉN. Copyright 2015.



# 1 [Práticas]

## 2

### 3

#### 4

##### 5

###### 6

###### 7

###### 8

###### 9

###### 10

###### 11

###### 12

###### 13

###### 14

- Visualize o Fluxo de Trabalho
- Limite o Trabalho em Progresso (WIP)
- Gerencie o Fluxo
- Deixe as Políticas do Processo Explícitas
- Implemente Ciclos de Feedbacks
- Melhore colaborativamente (usando modelos e o método científico)



1

# [Destaque]

2

3

4

- 5 • Visualize o Fluxo de Trabalho

- 6 • Limite o Trabalho em Progresso (WIP)

- 7 • Deixe as Políticas do Processo  
8 Explícitas

- 9  
10 • Implemente Ciclos de Feedbacks

11

12

13

14



1

2

3

- **Visualize o Fluxo de Trabalho:**

4

5

O próprio quadro Kanban representa o estado real do seu fluxo de trabalho com todos os seus riscos e especificações.

6

7

8

9

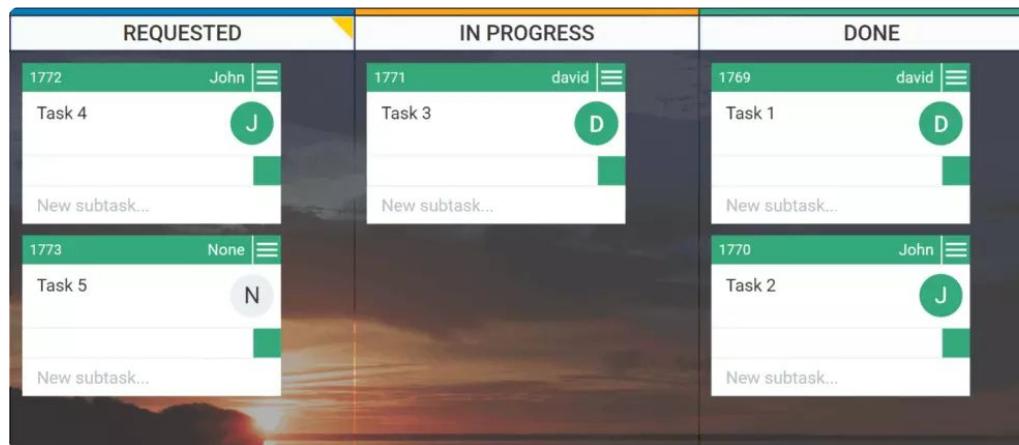
10

11

12

13

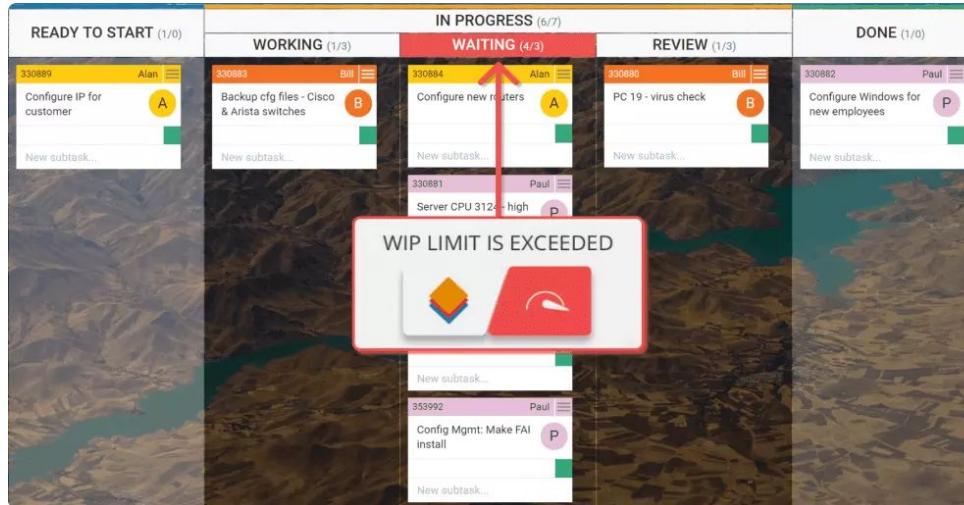
14



1

## • Limite o Trabalho em Progresso (WIP):

Uma das principais funções do Kanban é garantir que um número gerenciável de itens ativos esteja em andamento a qualquer momento. Se não houver limites de trabalho em andamento, você não está fazendo Kanban.



5

6

7

8

9

10

11

12

13

14

1

2     ● **Deixe as Políticas do Processo Explícitas:**

3     Você não pode melhorar algo que não entende. É por isso que seu processo  
4     deve ser claramente definido, publicado e socializado. As pessoas não se  
5     associariam e participariam de algo que não acreditam ser útil.

6

7

8

9

10

11

12

13

14



1

2     • **Implemente Ciclos de Feedbacks:**

3     Para equipes e empresas que desejam ser mais ágeis, implementar ciclos de  
4     feedback é uma etapa obrigatória. Eles garantem que as organizações estejam  
5     respondendo adequadamente às mudanças potenciais e permitem a transferência  
6     de conhecimento entre as partes interessadas.



10

11

12

13

14

1

- **Scrum vs. Kanban:**

2

3

4

5

6

7

8

9

10

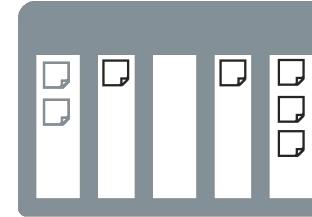
11

12

13

14

*A diferença mais importante entre Kanban e Scrum é que o primeiro é um método, enquanto o último é um framework. O Kanban constrói um modelo de entrega contínua onde as equipes liberam valor assim que estão prontas, enquanto o Scrum organiza o trabalho em Sprints.*



Kanban



**uniesp**

Centro Universitário