

# Kanban

**Disciplina: Engenharia de Software**

Gustavo Lacerda

[gustavo.lacerda@prof.una.br](mailto:gustavo.lacerda@prof.una.br)



---

# Kanban



# Fundamentos

---

- ▶ A palavra Kanban:
  - ▶ vem do japonês e significa **Cartão Visual**.
  - ▶ é utilizada para descrever o sistema que vem sendo utilizada há décadas pela Toyota para visualmente controlar e equilibrar a linha de produção.
  - ▶ vem sendo utilizado por mais de 50 anos no sistema de produção Lean na Toyota.



---

# Quando devo considerar trabalhar com o Kanban?



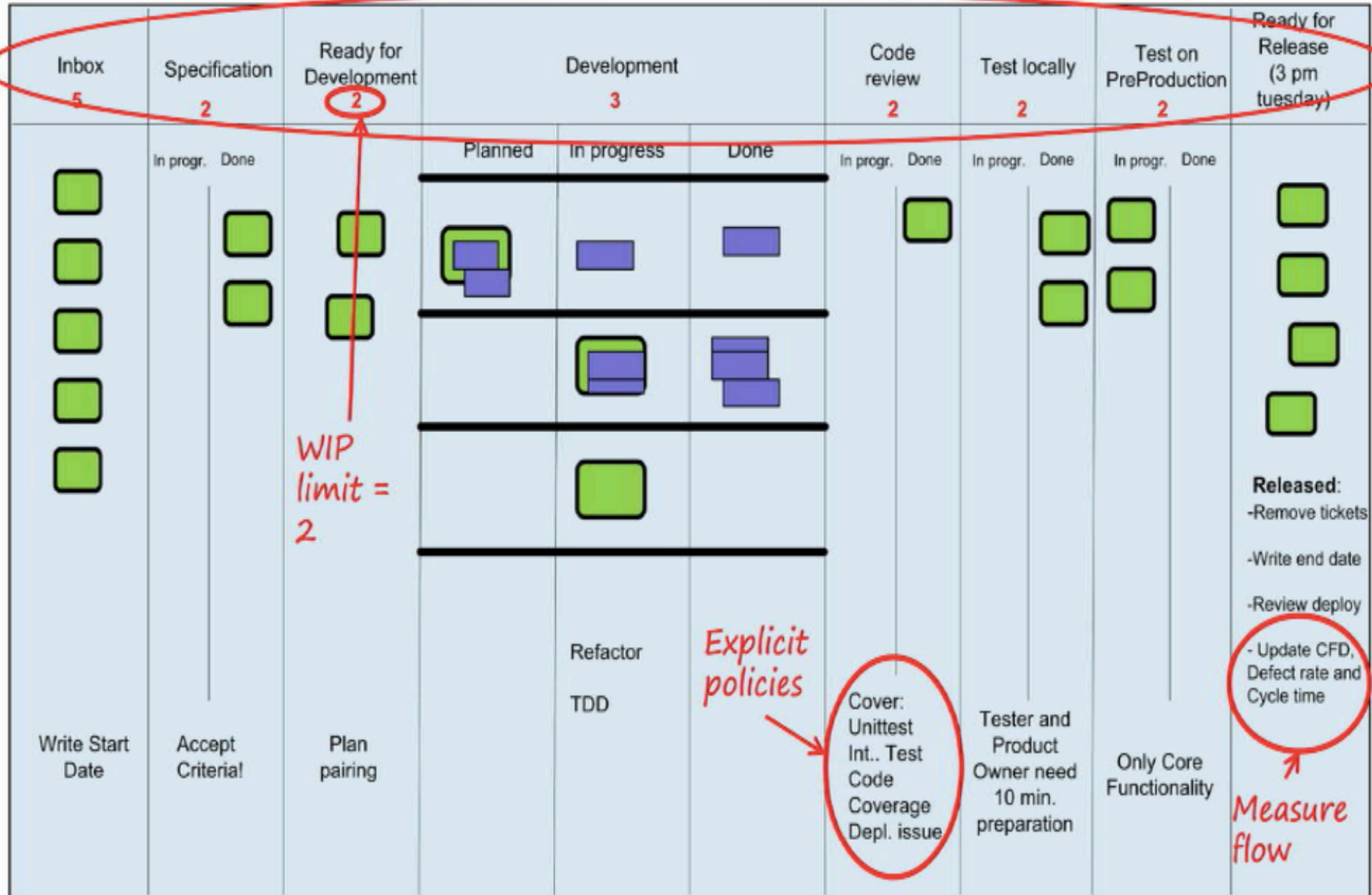
# O Que é o Kanban?

- ▶ Existem diversas abordagens para o Kanban, mas a maioria dos especialistas concorda que o Kanban é um método de gestão de mudanças, que dá ênfase as seguintes princípios:
  - ▶ Visualizar o trabalho em andamento (WIP)
  - ▶ Visualizar cada passo em sua cadeia de valor.



# O Que é o Kanban?

- ▶ Limitar o Trabalho em Progresso, restringindo o total de trabalho permitido para cada estágio
- ▶ Medir e gerenciar o fluxo, para poder tomar decisões bem embasadas, além de visualizar as consequências dessas decisões.
- ▶ Identificar oportunidades de melhorias, criando uma cultura, na qual a melhoria contínua é reponsabilidade de todos.





# O Que é o Kanban?

- ▶ O foco do Kanban é conduzir mudanças evolucionárias, e estes passos tem-se se mostrado extremamente úteis para esse objetivo.
- ▶ Ao visualizar o fluxo e estabelecer os limites de WIP, garantimos que **nunca** se pode introduzir mais trabalho no sistema que a capacidade do sistema de processar esse trabalho.





# O Que é o Kanban?

- ▶ Está disponível apenas uma quantidade limitada de permissões de trabalho (kanbans). Então é necessário que se complete o trabalho existente antes que um novo trabalho possa ser iniciado.
- ▶ Isto resulta em funcionalidade sendo **puxadas** pelo sistema, com base na capacidade deste, em vez de **empurradas** com base em previsões ou demandas.



---

# Como começar?



# Mudanças evolucionárias



- ▶ O Kanban é construído sobre os conceitos da mudança evolucionária. Uma abordagem é começar entender como funciona atualmente seu sistema de entrega de software.
- ▶ Quando conseguir visualizar medir e gerenciar o fluxo que esta sendo utilizado, melhore um passo de cada vez.
  - ▶ Aliviando seu maior **gargalo**.



# Mitos e Fatos

---

- ▶ **MITO:** O Kanban é adequado apenas para equipes que trabalham com tarefas pequenas e padronizadas, como as de operações e manutenção.
- ▶ **FATO:** O Kanban é altamente inspirado no desenvolvimento Lean de Produtos, e se provou ser uma opção para o desenvolvimento quanto para manutenção de software.



# Mitos e Fatos

---

- ▶ **MITO:** O Kanban e o Scrum são opostos
- ▶ **FATO:** Nenhum dos princípios do Kanban restringe o uso do Scrum. Nada impede de se começar com o Scrum e utilizar o Kanban para impulsionar futuras mudanças.



# Mitos e Fatos

---

- ▶ **MITO:** As equipes que utilizam Kanban não utilizam período fixos de tempo (timeboxes)
- ▶ **FATO:** Os timeboxes não são exigidos no Kanban, mas devem ser utilizados quando ajudarem a otimizar o fluxo , o nível de feedback e a qualidade.



# Mitos e Fatos

- ▶ **MITO:** As equipes Kanban não fazem estimativas
- ▶ **FATO:** As estimativas não obrigatórias, mas devem ser usadas quando apropriadas. A maioria dos projetos Kanban de desenvolvimento usam algum grau de dimensionamento inicial, ou para assegurar o gerenciamento, a priorização e o alinhamento ideal de portfólio de projetos.



---

# Passo a Passo







# **Passo 1.**

# **Visualizar o Fluxo de Trabalho**



# Passo 1: Visualizar o Fluxo de Trabalho

- ▶ O primeiro passo para visualizar o seu fluxo de trabalho é entender como seu sistema **atual funciona**.
  - ▶ é importante resistir a tentação de realizar mudanças.
  - ▶ deve-se mapear **todo o fluxo de trabalho** para entrega do software, sem focar apenas na parte do "desenvolvimento"
- ▶ Existem algumas formas de se mapear o fluxo de trabalho, utilizaremos o **Mapa de Cadeia de Valor**



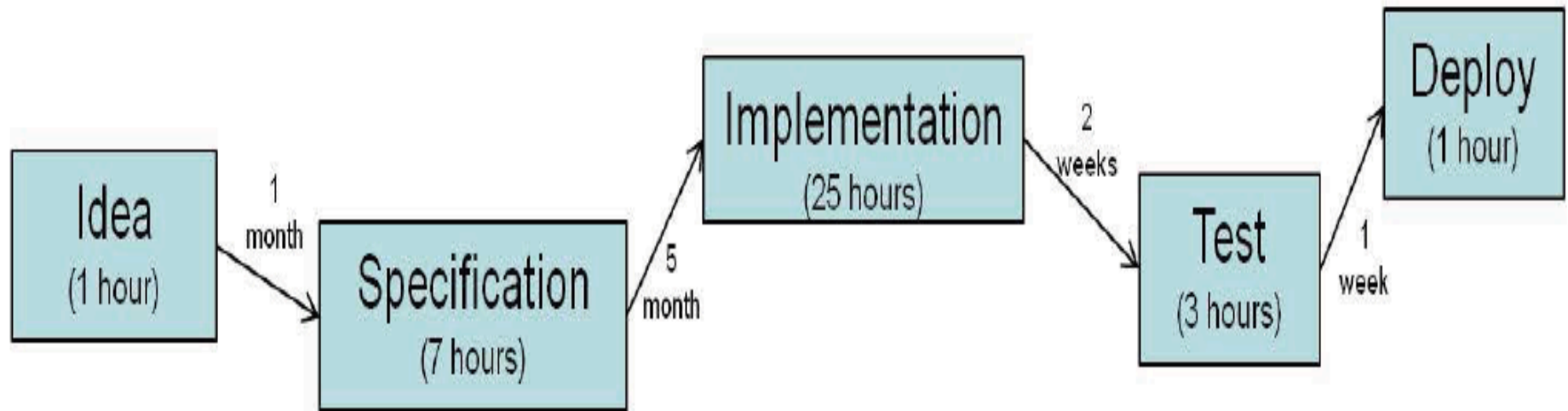
# Mapa de Cadeia de Valor

---

- ▶ é uma visualização dos estágios pelos quais passa o trabalho, desde a matéria prima até o produto final - em *software*, de uma ideia vaga até a funcionalidade implantada em produção.
  - ▶ O mais importante é pensar em cada estágio como sendo a representação de um grupo de atividades.



# Mapa de Cadeia de Valor





# Visualizar o seu sistema de entregas

PO Inbox	Specification	Ready for development	Development			Code review	Test with PO and tester	Ready for release	Release
			Planned	In progress	Done				

Activity  
Stage

Buffer  
Stage



# Visualização do Trabalho

- ▶ Caso visualizar seu trabalho seja difícil, **PARE**.
- ▶ Não vá além, a não ser que tenha conseguido visualizar o trabalho que você faz.
- ▶ Regra geral: é possível gerenciar apenas o trabalho que se poder ver.
- ▶ Visualizar o trabalho é **difícil** na vida real.
  - ▶ As pessoas sabem que estão fazendo coisas que não deveriam
  - ▶ Medo de serem punidas caso seus superiores saibam como as coisas realmente funcionam.



# Visualização do Trabalho

---

- ▶ **Benefícios:**
  - ▶ **Foco no todo:** Fica mais visível exatamente como o seu trabalho afeta as outras pessoas e vice-versa
  - ▶ **Transparência:** Todos sabem exatamente o que está acontecendo e nenhuma informação é escondida
  - ▶ **Identificação de desperdícios:** Surgirá naturalmente um questionamento da razão pela qual as coisas são feitas.



---

# **Passo 2.**

# **Limitar o WIP**







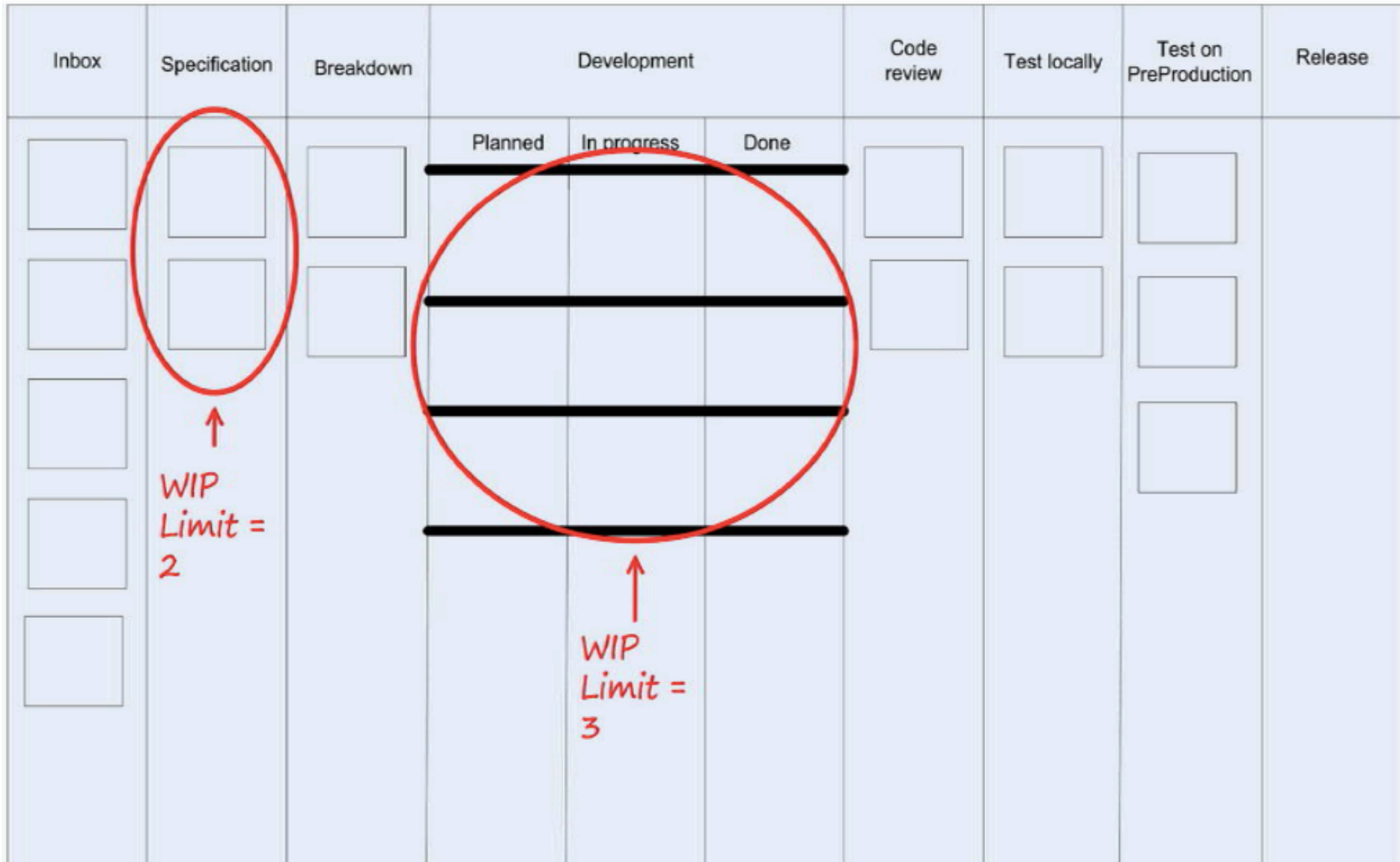
## Passo 2: Limitar o WIP

---

- ▶ **WIP:** descreve o total de trabalho em progresso no Sistema Kanban.
- ▶ **Tempo de ciclo:** é o tempo que se leva para um item de trabalho passar pelo nosso sistema Kanban.(Implementação até Produção)
- ▶ **Produção por unidade de tempo:** é a média do número de itens produzidos em um determinado período de tempo.



# Visualizando Prazos de WIP





# Identificação dos Prazos Ideais

---

- ▶ Em qualquer situação, é importante que se determine uma política explícita sobre como serão tomadas as decisões para alterar um prazo.
- ▶ Para maximizar o aprendizado, o ideal é tomar essas decisões em conjunto com a equipe, garantindo que todos possam expor suas opiniões e entender as decisões.
- ▶ Não se trata apenas do fluxo, mas também de aprendizado.



---

# **Passo 3.**

## **Estabelecer políticas explícitas para garantia de qualidade**



# Porque a qualidade é tão importante?

---

- ▶ No Kanban, concentra-se em criar políticas explícitas que otimizem a qualidade e deem consistência ao sistema de entrega de *software*.
- ▶ Essas políticas são utilizadas como base para a melhoria contínua.



# Entendendo a Qualidade

---

- ▶ Uma interface de baixa qualidade que impedem a conclusão de uma tarefa, ou um defeito que atrapalha o fluxo de trabalho – ambos são problemas que “stressam” o sistema Kanban e qualquer outro tipo de metodologia.
- ▶ Ciclos de desperdício.
- ▶ **As vezes** é aceitável lançar um produto com baixa qualidade para se obter mais feedback
- ▶ **Na maioria dos casos**, porém, a baixa qualidade se trata apenas de um sintoma de processo imaturo.



# Visualização das Políticas

Inbox 5	Specification 2	Ready for Development 2	Development 3			Code review 2	Test locally 2	Test on PreProduction 2	Ready for Release (3 pm tuesday)
	In progr. Done		Planned	In progress	Done	In progr. Done	In progr. Done	In progr. Done	
Write Start Date	Accept Criterial	Plan pairing				Cover: Unittest Int.. Test Code Coverage Depl. issue	Tester and Product Owner need 10 min. preparation	Only Core Functionality	<b>Released:</b> -Remove tickets -Write end date -Review deploy - Update CFD, Defect rate and Cycle time
				Refactor TDD					



# Entendendo a Qualidade

- ▶ Estágios inteiros podem ter políticas para garantia de qualidade, e que políticas também podem servir para a garantia da consistência e de qualidade do próprio processo.
- ▶ No começo, pode parecer penoso analisar e aprender com cada defeito que surgir. Mas com o tempo, conforme a qualidade melhora, o processo se tornará mais natural.
  - ▶ TOLERANCIA ZERO: não significa que nunca mais possam existir defeitos, e sim que devemos conscientemente refletir sobre cada defeito e buscar a perfeição.





---

# **Passo 4.**

# **Ajustar o ritmo de entrega**





# Entendendo o Ritmo de Entrega

---

- ▶ Encontrar o ritmo de trabalho ideal é uma das tarefas mais importantes no Desenvolvimento de Produtos, pois ajuda otimizar os ciclos de feedback, reduzir os riscos e otimizar o seu processo de entrega.
- ▶ Quanto mais funcionalidade são incluídas em uma entrega, **mais barato** será **o custo** por funcionalidade.
- ▶ Em contrapartida o custo de espera é maior, uma vez que cada funcionalidade terá de esperar mais tempo para ser lançada. Isso resulta em perda de valor de negócio, *feedback* desatualizado, decisões e menos envolvimento com o usuário.



# Estabelecendo os Ritmos Ideais

- ▶ O movimento de implantação contínua tem mostrado que é possível lançar versões confiáveis para produção várias vezes ao dia, para sistemas lidando com milhões de pessoas.
- ▶ Isto pode ser atingido com um procedimento completamente automatizado de lançamento, e todo um conjunto de testes unitários, testes de integração e de regressão
  - ▶ <https://builds.apache.org/>



---

# **Passo 5.**

# **Medir o fluxo de trabalho**





# Entendendo as Métricas

---

- ▶ é comum ver métricas sendo usadas para tornar os **GPs** responsável por aspectos sobre os quais não se tem controle.
- ▶ Critérios de sucessos fixos, estabelecidos quando nem se conhecia o sistema a ser desenvolvido.
- ▶ **GPs** orgulhosos de terem feito suas equipes trabalharem além do horário por três meses, ou que, por algum esforço heroico, conseguiram controlar as coisas no último momento, depois de semanas de caos.



# Entendendo as Métricas

---

- ▶ **Boa regra:** seu sistema nunca terá mais capacidade do que a capacidade que já provou ter.
- ▶ Seguindo essa regra, pode-se evitar o excesso de otimismo e comparações com o sucesso dos outros.



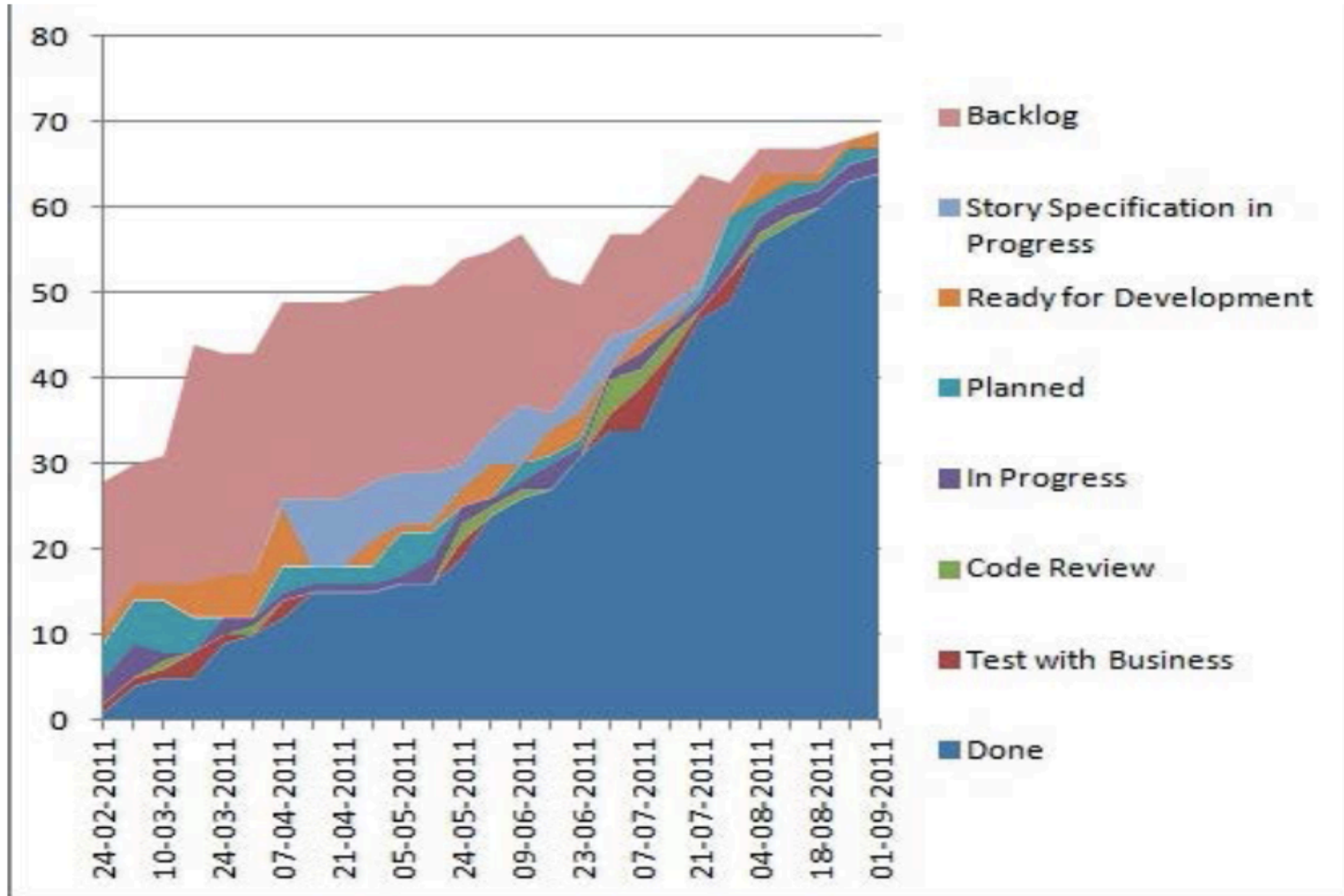
# O que medir?

---

- ▶ Diagramas de fluxo cumulativo (DFC)
  - ▶ Diagramas de fluxo cumulativo mostram essencialmente um retrato da quantidade de trabalho em seu sistema, para cada estágio de sua cadeia de valor, no decorrer do tempo.



# Diagrama de Fluxo Cumulativo







---

# Passo 6. Priorização





# Custo de Atraso

- ▶ Um conceito importante usado para a priorização é o Custo de Atraso (COD).
- ▶ Representa a receita(perdida) ou a economia(ganha) com a escolha de não se trabalhar em um determinada requisito. A mais alta prioridade deve ser o item com o maior COD.
- ▶ Aprender a associar um valor econômico ao trabalho desempenhado é um exercício de amadurecimento para projeto e organizações.



# Visualizando Prioridades

- ▶ **Riscos e incertezas:** obtenha informações o mais cedo possível, para apoiar decisões de alto risco e de alto impacto;
- ▶ **Necessidade básicas:** infraestrutura do projeto;
- ▶ **Tamanho equilibrado:** misture histórias de diferentes tamanhos para manter um fluxo constante;
- ▶ **Tipos de história equilibrados:** misture histórias funcionais e não-funcionais para garantir um fluxo constante de valor
- ▶ **Dependências:** tratar dependências pro ativamente, para que o trabalho não fique impedido



---

# **Passo 7.**

## **Identificação de Classes de Serviço**



# Identificando Tipos de Trabalho

---

- ▶ Um bom começo é identificar os diferentes tipos de trabalho:
- ▶ Casos de Uso(Pequeno, Médio e Grande)
- ▶ Bugs(Críticos, Impeditivos, Melhorias)
- ▶ Tarefas de Suporte
- ▶ Implantação



# Classes de Serviço

---

## ▶ **Classe Padrão**

- ▶ Custo Adicional: R\$0,00
- ▶ Tipo de trabalho: Melhorais, Casos de Uso
- ▶ Tratamento especial: Nenhum

## ▶ **Classe Prioritária**

- ▶ Custo Adicional: R\$500,00
- ▶ Tipos de trabalho: Defeitos críticos, Casos de Uso de alta prioridade
- ▶ Tratamento especial: Ganha prioridade em todos os estágios



# Classes de Serviço

---

## ▶ Classe de Prazo Fixo

- ▶ Custo Adicional: R\$0,00 até R\$2000,00
- ▶ Tipos de Trabalho: Casos de Uso
- ▶ Tratamento especial: tem prioridade em todos os estágios, se o prazo estiver próximo; caso contrário é tratada como uma classe de serviço padrão, A implantação emergencial pode ser necessária.



# Classes de Serviço

---

## ▶ Classe Urgente

- ▶ Custo adicional: R\$3000,00 até R\$5000,00
- ▶ Tipo de Trabalho: Defeito Impeditivo
- ▶ Tratamento Especial: Romper os limites de WIP, para todo o WIP existente; implantação emergencial





# **Passo 8.**

## **Gerenciamento do Fluxo**





# Gerenciamento do Fluxo

---

- ▶ Ao chegar neste ponto, já estaremos operando em um ambiente ágil e maduro. Visualizamos o fluxo de trabalho, o trabalho em progresso está limitado, políticas de qualidade estão estabelecidas, o fluxo já esta sendo medido.
- ▶ O próximo passo é aprender a interpretar o sistema e tomar ações apropriadas quando houver oportunidade de melhoria.



# Otimizando o fluxo, não a utilização

- ▶ Ao procurar oportunidades de melhorias, evite pensar em termos de utilização. Busque oportunidade para aumentar o fluxo de itens de trabalho passando pelo seu sistema e faça perguntas como:
  - ▶ Os WIP são adequados?
  - ▶ É possível diminuir o tamanho dos casos de uso?
  - ▶ Existe alguma forma de identificar funcionalidades grandes, antes que entrem em seu sistema e acabem bloqueando a capacidade por longos períodos?



# Otimizando o fluxo, não a utilização

---

- ▶ é possível equilibrar o tamanho das histórias de usuário para criar um fluxo mais contínuo?
- ▶ Pode-se treinar a equipe visando a flexibilidade, para evitar gargalos?
- ▶ Existem buffers adequados em seu sistema para lidar com variações?
- ▶ O foco está na otimização do todo e não na otimização de estágios individuais?



# Introduzindo Pulmões

---

- ▶ Se você sabe que existe um gargalo no sistema, é uma boa ideia protegê-lo com um pulmão logo no início.
  - ▶ Escolher o tamanho correto para o pulmão exige experiência
  - ▶ Se o Pulmão esvazia com frequência, é recomendável aumentar o seu tamanho e avaliar o problema.



# Introduzindo Pulmões

- ▶ Se você sabe que existe um gargalo no sistema, é uma boa ideia protegê-lo com um pulmão logo no início.
  - ▶ Escolher o tamanho correto para o pulmão exige experiência
  - ▶ Se o Pulmão esvazia com frequência, é recomendável aumentar o seu tamanho e avaliar o problema.



# Planejando Entregas

---

- ▶ Projetos são desenvolvidos sob as restrições:
  - ▶ Custo
  - ▶ Tempo
  - ▶ Escopo
- ▶ Normalmente não se pode fixar as três restrições ao mesmo tempo. Melhor saída **escopo flexível**.
- ▶ Atrasar uma data combinada é difícil e significa muito tempo gasto em reorganização, coordenação e comunicação
- ▶ Aumentar o pessoal raramente é uma ideia para atingir um prazo apertado



# Planejando Entregas

---

- ▶ Para trabalhar com o escopo flexível deve existir tanto uma disciplina quanto a habilidade de medir o progresso com precisão, garantindo que as decisões são bem fundamentadas em um ambiente de mudanças contínuas.





---

# **Passo 9.**

## **Estabelecer SLAs**





# Estabelecer SLAs

---

- ▶ é o momento de mostrar os resultados para o público externo.
- ▶ Possuir um sistema estável, e utilizar métricas simples para o medir o desempenho do sistema, torna possível estabelecer acordos de níveis de serviços (SLAs) que realmente são cumpridos.
- ▶ Evita a volta ao caos e o apagamento de incêndios.
- ▶ Os sistemas Kanban, são orientados pelo fluxo.



# SLAs

---

## ▶ **Classe Padrão**

- ▶ Média: 15 dias
- ▶ 90% dentro do prazo de 21 dias
- ▶ Todas no prazo máximo de 30 dias

## ▶ **Classe urgente**

- ▶ Média: 2 dias
- ▶ 90% dentro do prazo de 3 dias
- ▶ Todas no prazo máximo de 4 dias

## ▶ **Classe de prazo fixo**

- ▶ 98% dentro do prazo



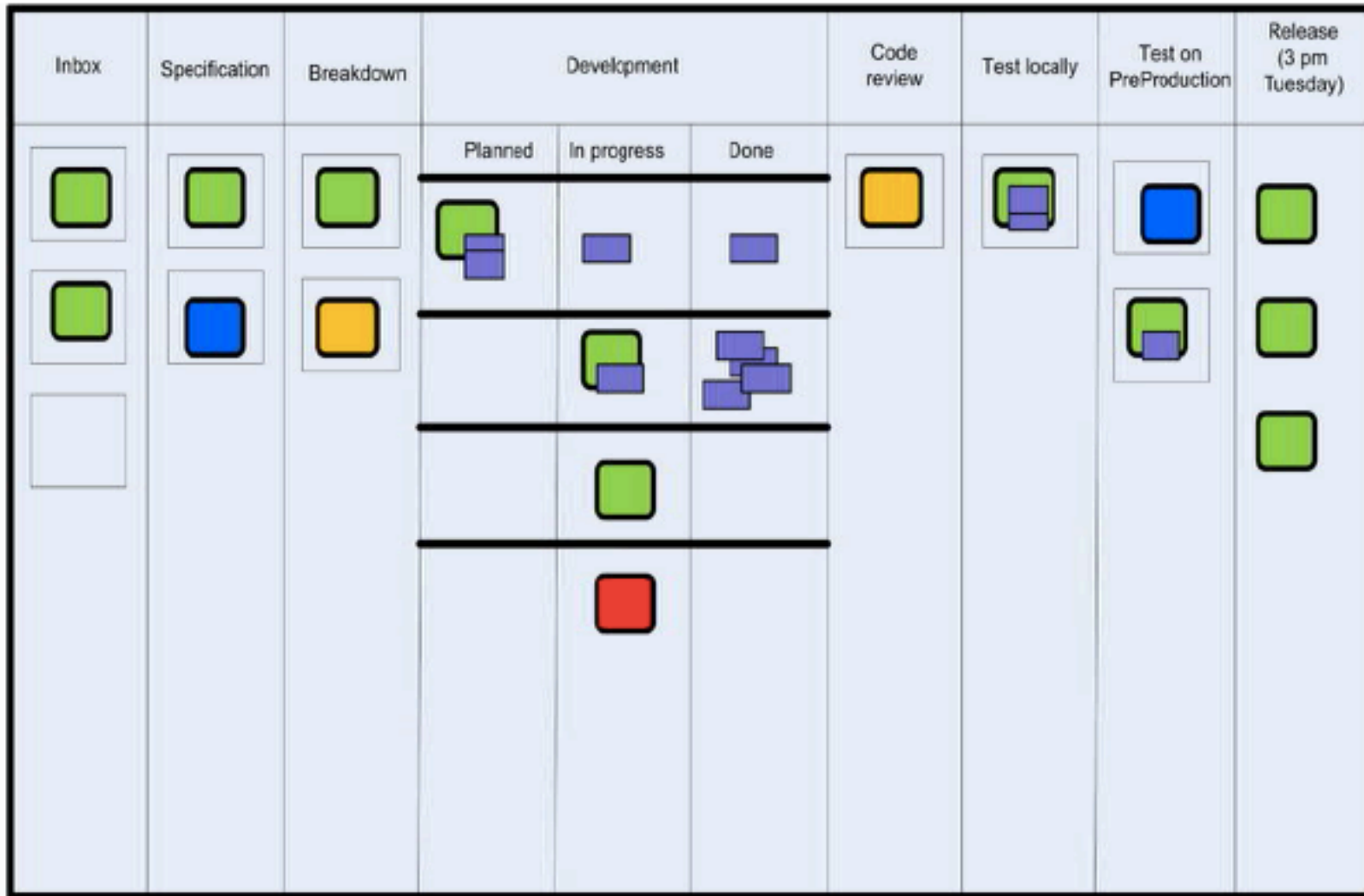
# SLAs

---

- ▶ **Classe prioritária**

- ▶ Média: 8 dias
  - ▶ 90% dentro do prazo de 13 dias
  - ▶ Todas no prazo máximo de 18 dias
- ▶ O fundamental é que os números são baseados em informações colhidas por meio do acompanhamento do desempenho de nosso sistema de entregas.
- ▶ Para muitos clientes, essas informações são valiosas para priorização. Saber que esses números são reais gera confiança e colaboração.

# SLAs





---

# **Passo 10.**

# **Melhoria Contínua**





# Melhoria Contínua

---

- ▶ Criar e manter um ciclo constante de melhorias é um dos elementos mais importantes ao implementar o Kanban.
- ▶ A distribuição de informações com a visualização do fluxo de trabalho e de políticas explícitas, além dos SLAs para cada classe de serviço, é grande incentivo ao diálogo constante sobre oportunidade de melhoria - muito além do que se vê em projetos de software tradicionais.



# Melhoria Contínua

---

- ▶ A grande distribuição de informações com a visualização do fluxo de trabalho e de políticas explícitas, além dos SLAs para cada classe de serviço, é grande incentivo ao diálogo constante sobre oportunidade de melhoria - muito além do que se vê em projetos de software tradicionais.
- ▶ Para ser bem sucedido com o Kanban, é necessário o comprometimento da gestão. Além disso, as pessoas que estão adotando os princípios em seus trabalhos diários devem entender sempre porque isso faz sentido.