



WACAD018 – Processo de Desenvolvimento Ágil - Aula 03

Júlia Luiza

jlslc@icomp.ufam.edu.br

Cronograma: Aula 03

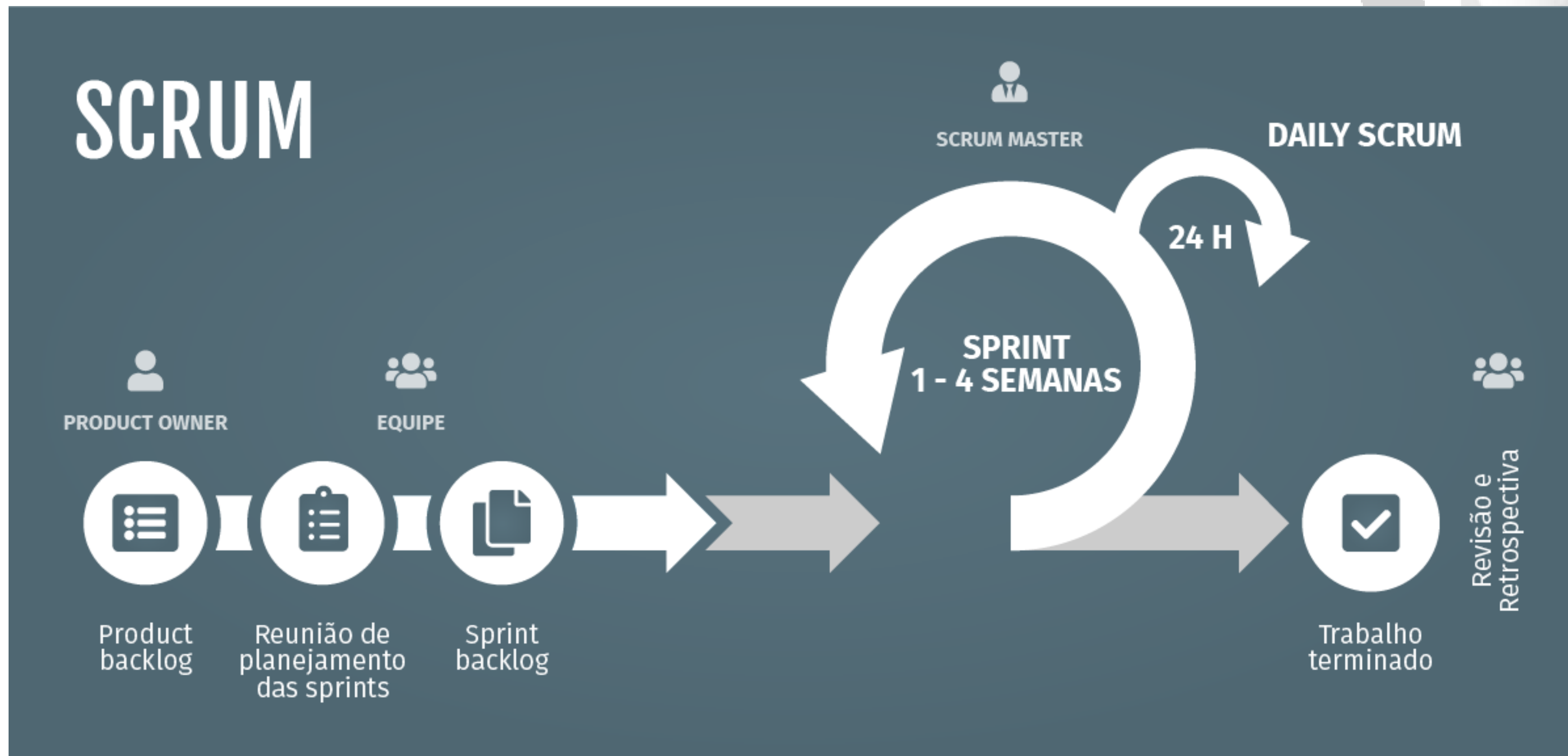
- **Scrum Parte 3:**
 - Velocidade da equipe;
 - Gráficos de Burndown;
 - Gráficos de BurnUp;
 - Benefícios e Desafios.
- **Kanban:**
 - O que é, como surgiu;
 - Princípios e como funciona;
 - Quadro Kanban;
 - Limites de trabalho em andamento;
 - Benefícios e Desafios.
- **Prática 03:** Construindo um quadro Kanban;



Métodos Ágeis: SCRUM

Parte 3 (Final)

Visão Geral do Processo



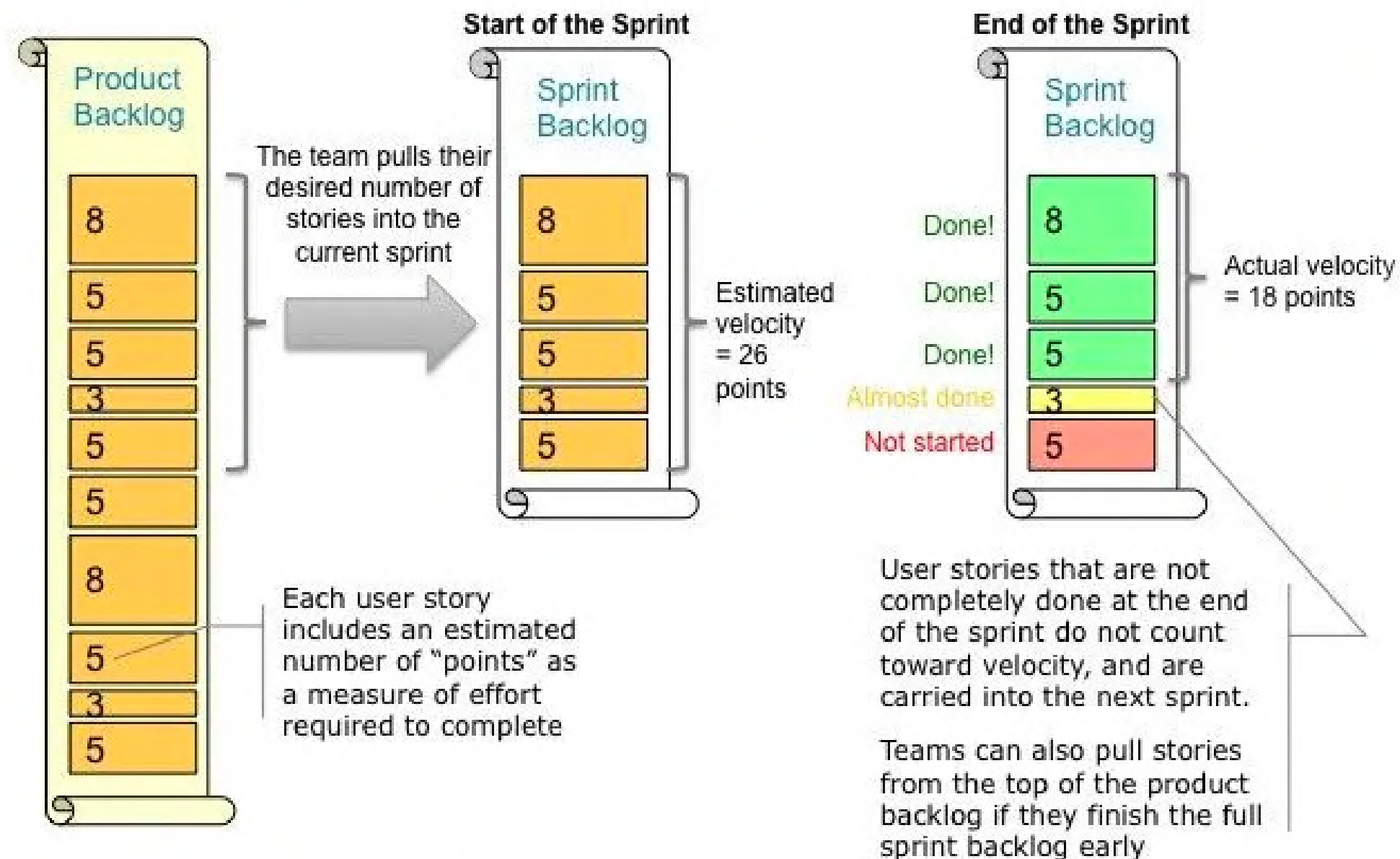
Velocidade da Equipe

- No contexto do Scrum, é uma **medida da quantidade de trabalho completada em um Sprint**;
- Sendo que a unidade de medida geralmente é em **pontos de história, horas ou itens de backlog**;
 - Os pontos de histórias parcialmente concluídas ou incompletas não devem ser contabilizados no cálculo da velocidade;
 - Atualmente, a própria organização do Scrum não incentiva o uso de 'horas' para estimar histórias e avaliar velocidade de equipes.
- **Auxilia na previsão de entregas futuras**;
 - Para fins de previsão, a média da velocidade dos três últimos Sprints deve ser usada;
 - Conhecendo a Velocidade, um PO pode descobrir quantos Sprints serão necessários para a equipe atingir um nível desejado de funcionalidade.



Velocidade da Equipe

"Velocity" is the Key Metric in Scrum



Velocidade da Equipe



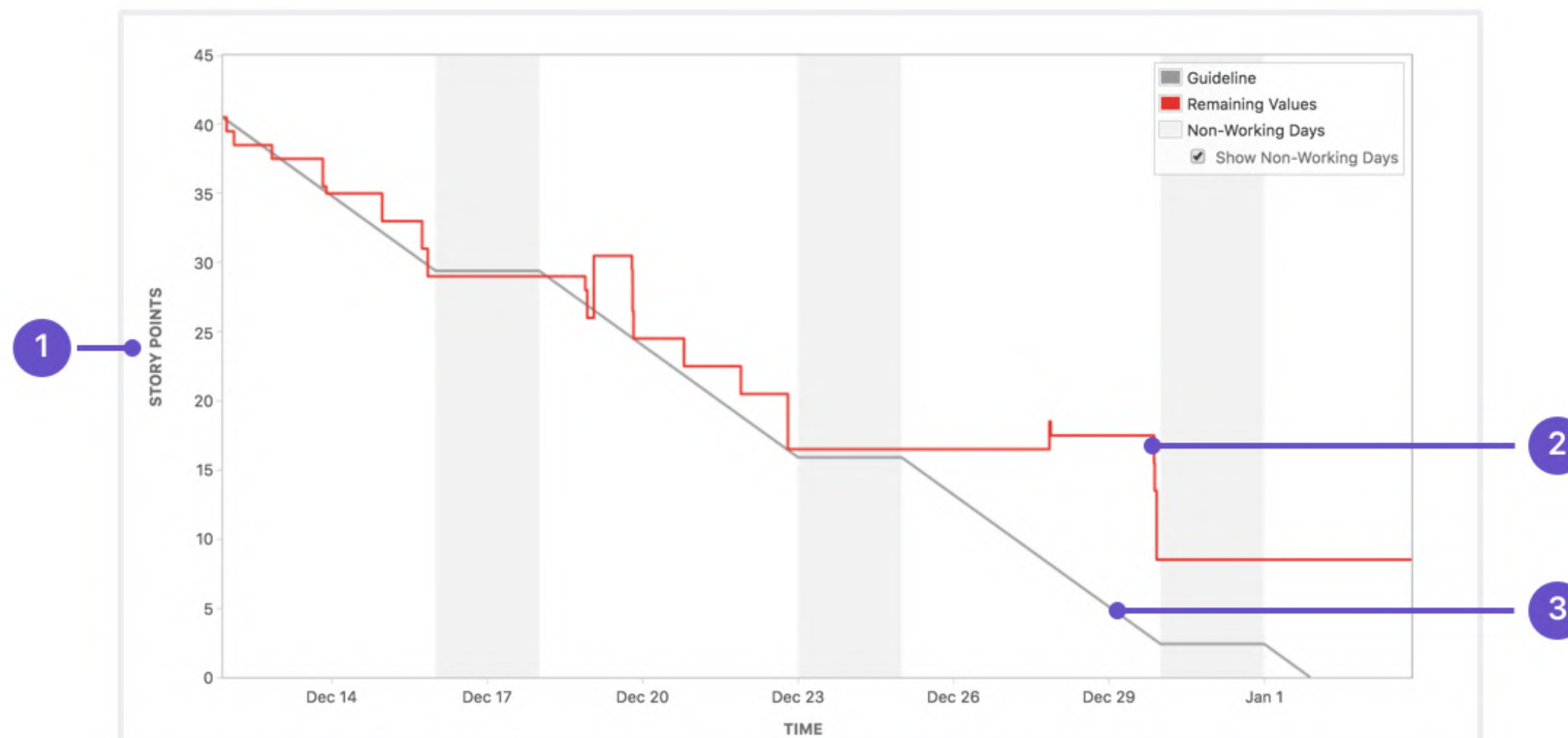
Velocidade da Equipe

- Com o tempo e maturidade da equipe, é esperado que a **velocidade seja cada vez mais estável** e vá melhorando em torno de 10% a cada sprint;
- Contudo, diversos fatores podem fazer com que a **velocidade varie de uma sprint para outra**, como por exemplo falta ou adição de membros;
- É importante entender a velocidade da equipe como não apenas uma métrica de produtividade, mas uma ferramenta importante para **previsão e melhoria contínua**.

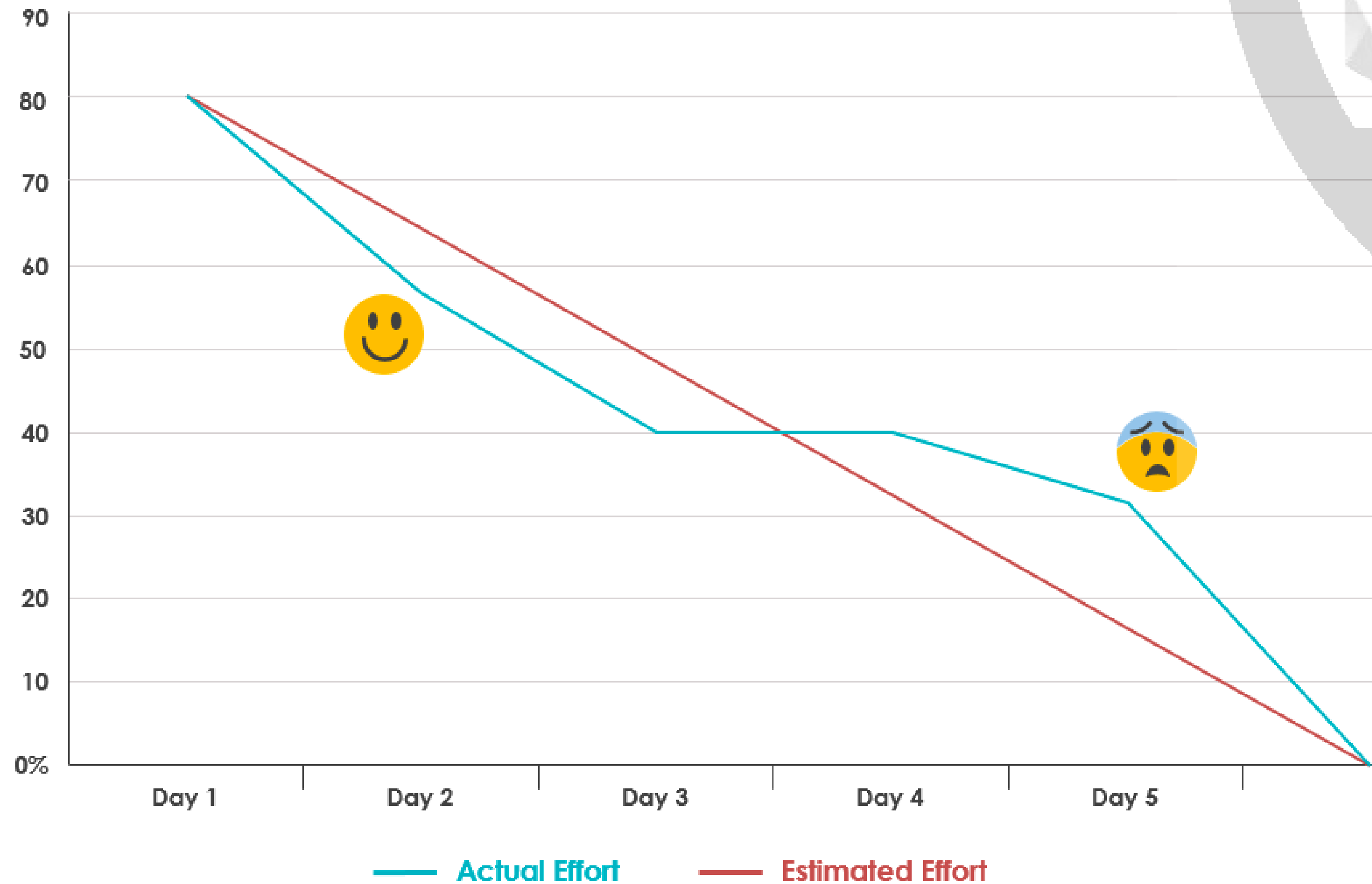


Gráficos de Burndown

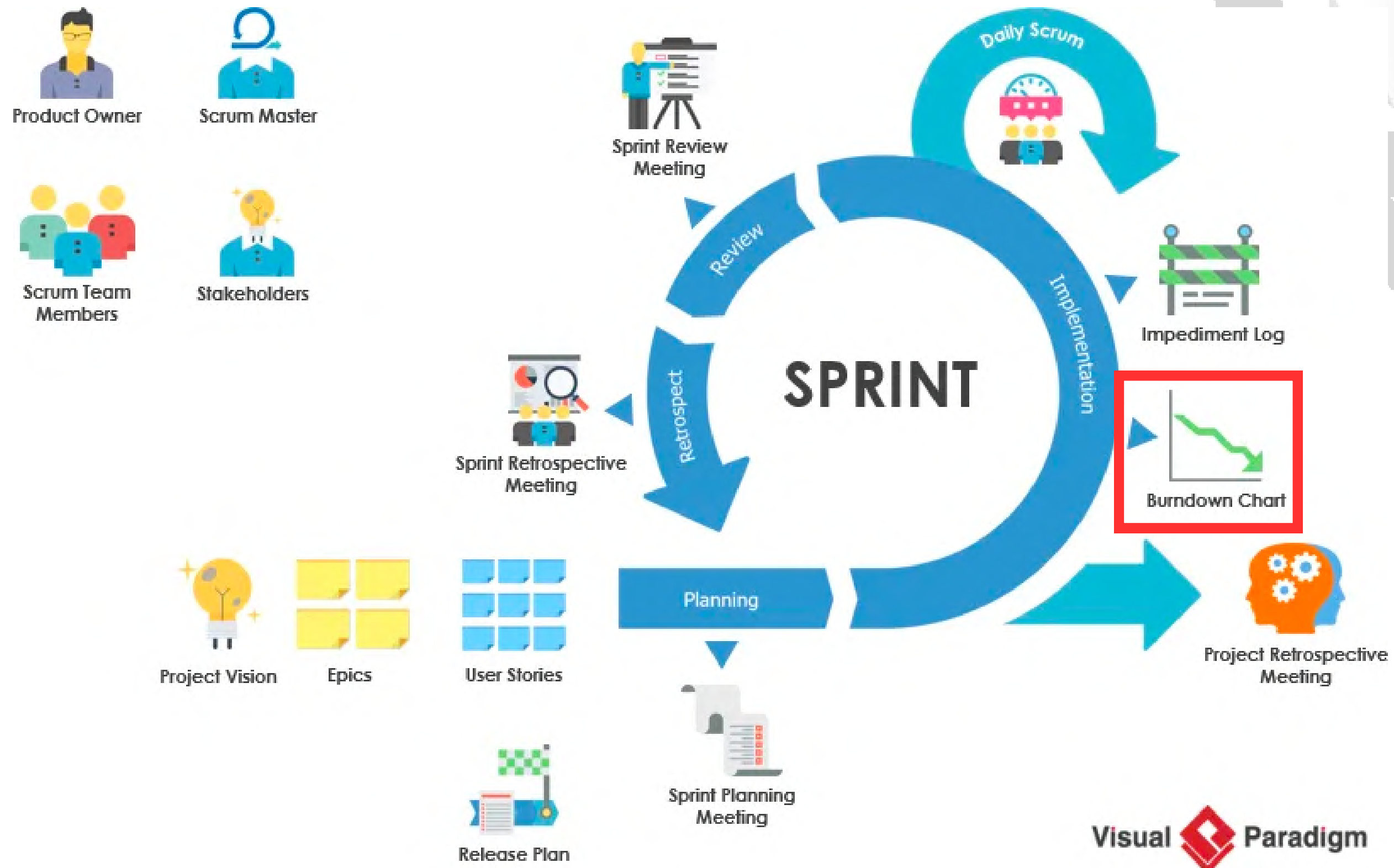
- Representam o **trabalho restante ao longo do tempo durante um sprint**, e mostram se a equipe está no caminho para concluir todas as tarefas do sprint;
- **Sua estrutura inclui:**
 - Eixo horizontal: Tempo (dias, horas, etc.).
 - (1) Eixo vertical: Trabalho restante (horas, pontos, etc.).
 - (2) Linha real: Representa a quantidade total de trabalho restante no sprint.
 - (3) Linha ideal: Mostra uma aproximação de onde sua equipe deveria estar, assumindo um progresso linear.



Gráficos de Burndown



Gráficos de Burndown



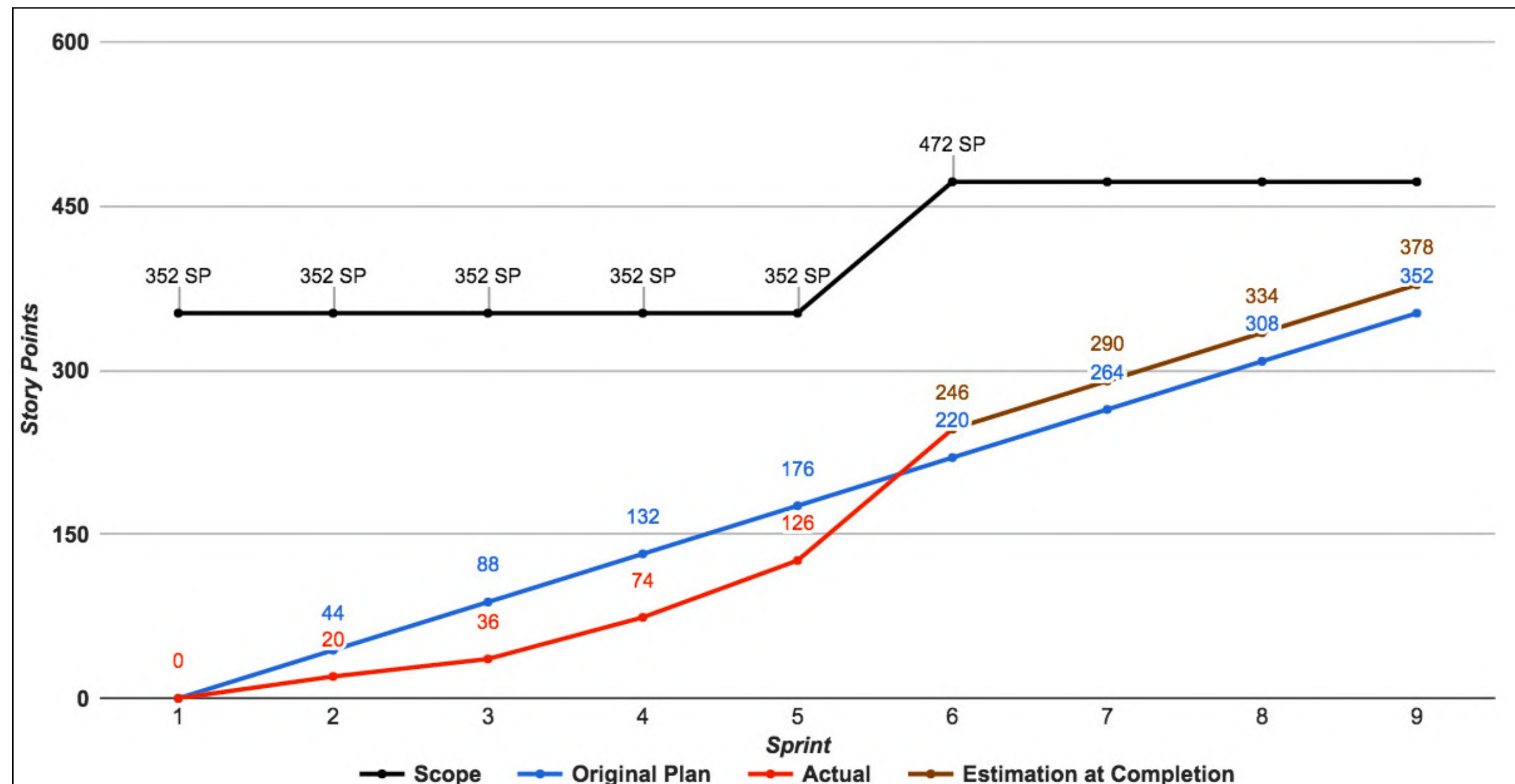
Gráficos de Burndown

- Usualmente, softwares de gestão de times ágeis, como o Jira, geram automaticamente esses tipos de gráficos;
 - Caso contrário, é possível construí-lo 'manualmente', sendo então importante mantê-lo atualizado diariamente para refletir o progresso real;
- Dessa forma, **cabe ao PO e ao time acompanhar diariamente o gráfico** para ter uma visão clara do progresso diária e de possíveis gargalos que estejam acontecendo;
 - Um gráfico de Burndown com a linha real muito acima da ideal, por exemplo, pode indicar que o time comprometeu-se com tasks acima de sua capacidade, ou que houve uma **falha durante a etapa de refinamento e/ou planejamento**.
- Além disso, é possível também utilizar o gráfico como **ferramenta de comunicação com stakeholders**.



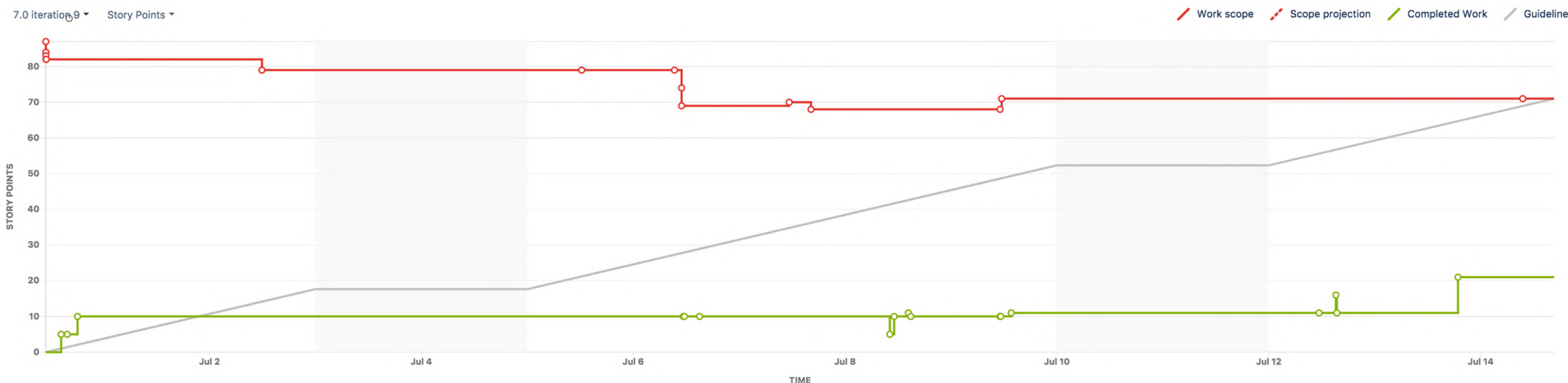
Gráficos de Burnup

- Serve para **visualizar o trabalho planejado, trabalho concluído e escopo total do projeto ao longo do tempo**;
 - Tornando possível identificar **lacunas entre planejamento e execução**;
 - E auxiliando na previsão de conclusão com base na tendência.
- Sua **estrutura** inclui:
 - Eixo horizontal: Tempo (dias, horas, etc.).
 - Eixo vertical: Trabalho restante (horas, pontos, etc.).

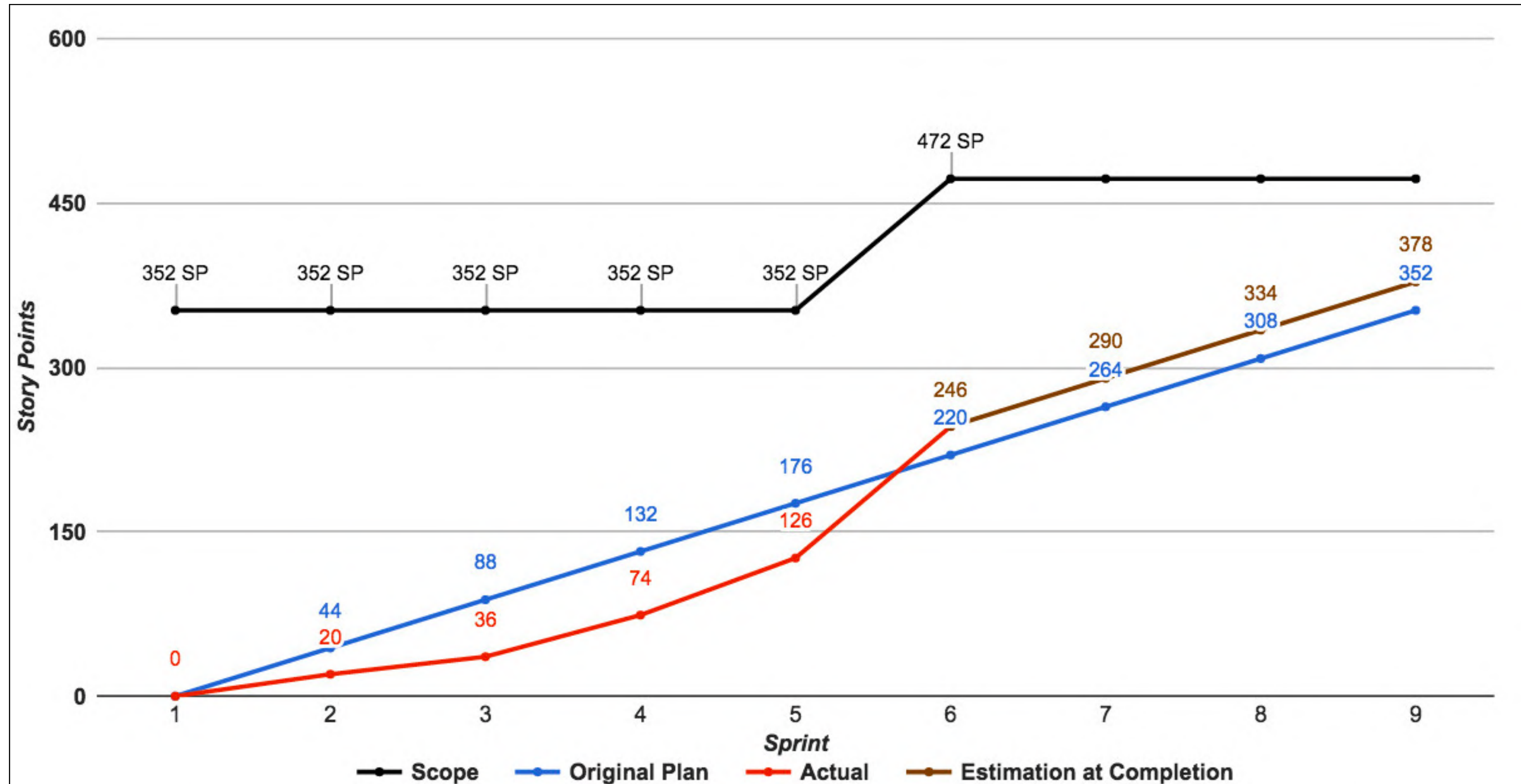


Gráficos de Burnup

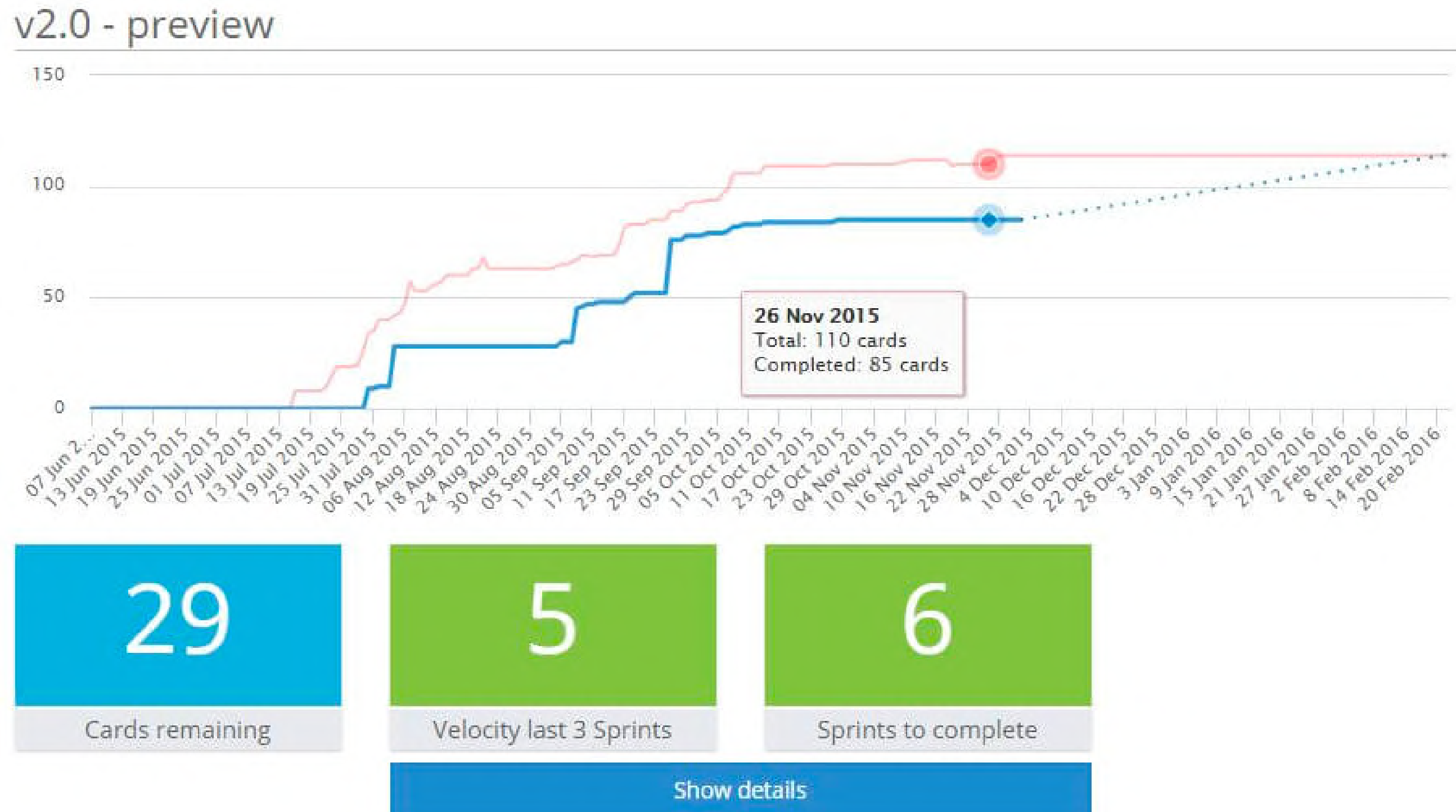
- Quanto as suas linhas, funciona da seguinte forma:
 - O **escopo do trabalho (a linha vermelha)** mostra o **trabalho planejado para o sprint**. Se essa linha aumentar ao longo do sprint, é um sinal de aumento de escopo.
 - O **trabalho concluído (a linha verde)** mostra o **trabalho que foi movido para 'Concluído'**.
 - A **diretriz (a linha cinza)** mostra **uma aproximação de como sua equipe deve progredir** ao longo do sprint para terminar no prazo.
 - A distância entre o Escopo do trabalho e as linhas de trabalho Concluído no gráfico é a quantidade de trabalho restante.
 - Quando todo o trabalho do sprint estiver concluído, as linhas se encontrarão.



Gráficos de Burnup



Gráficos de Burnup

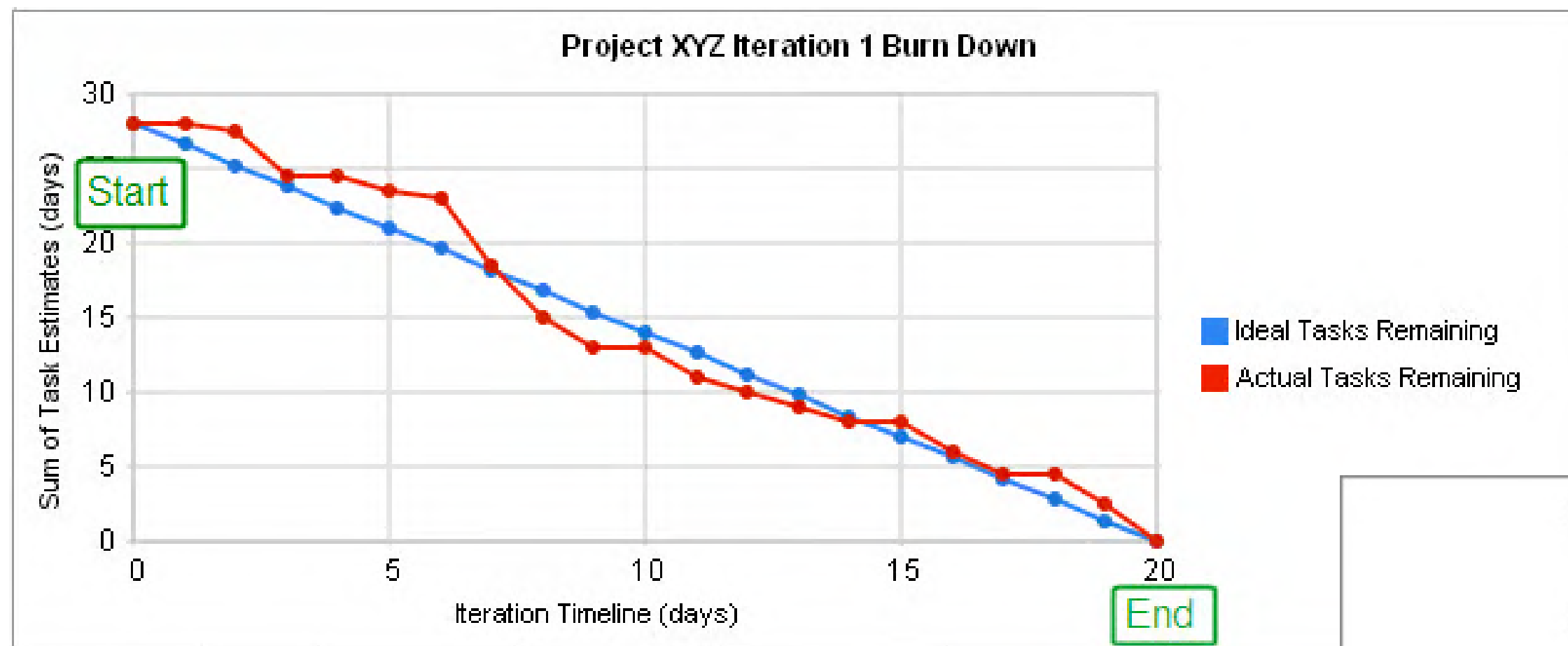


Gráficos de Burnup x Burndown

Para diferenciar:

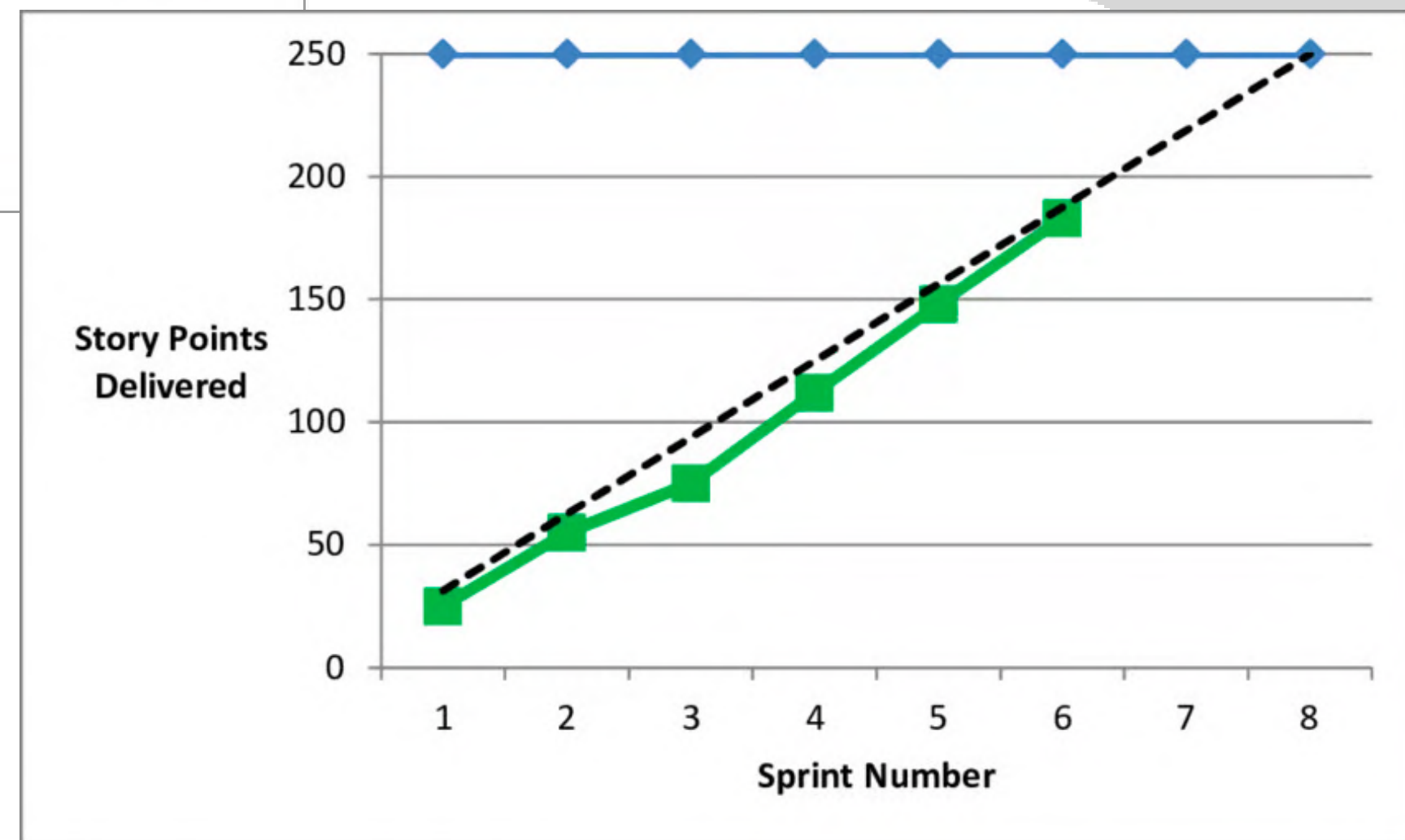
- **Burndown** ('queimar pra baixo'):
 - Mostra a **quantidade de trabalho restante na sprint**;
 - Ajuda a equipe a monitorar o **progresso diário** durante uma iteração específica;
 - **Não abrange toda a trajetória do projeto**, sendo mais útil para acompanhar o progresso de curto prazo e a projeção de entregas próximas.
- **Burnup** ('queimar pra cima'):
 - Mostra o **trabalho total e o trabalho concluído**;
 - Ajuda a equipe a entender não apenas o que foi concluído, mas também **como o escopo do projeto pode ter mudado** ao longo do tempo (por exemplo, alterações nos story points).
 - **Facilita a determinação** de marcos ou **datas aproximadas de conclusão** com base no trabalho concluído e no escopo total.

Gráficos de Burnup x Burndown



"um gráfico **burndown** é ótimo para enfatizar o que falta ser concluído, mas um gráfico **burnup** é melhor para ilustrar o trabalho que foi realizado."

<https://www.wrike.com/blog/what-is-a-burn-up-chart/>



Benefícios e Desafios

- **Benefícios do Scrum:**

- **Adaptabilidade:** Flexibilidade para lidar com mudanças durante o desenvolvimento do projeto.
- **Entrega Iterativa:** Possibilidade de entregas incrementais, possibilitando feedback constante.
- **Maior Colaboração:** Melhora na comunicação e cooperação entre os membros da equipe.
- **Transparência:** Processo claro e visível para todos os envolvidos no projeto.
- **Melhoria Contínua:** Através das retrospectivas regulares, permite identificar e resolver problemas de forma proativa, aumentando a eficiência ao longo do tempo.



Benefícios e Desafios

- **Desafios do Scrum:**

- **Resistência à Mudança:** Adaptação a um novo método de trabalho pode ser desafiador para equipes acostumadas com métodos tradicionais.
- **Requer Engajamento Total:** Todos os membros da equipe precisam se comprometer com os valores e práticas do Scrum.
- **Complexidade da Gestão de Mudanças:** Lidar com mudanças frequentes pode demandar uma gestão eficiente.
- **Maturidade da Equipe:** O Scrum funciona melhor em equipes maduras e autogerenciáveis.





Métodos Ágeis: Kanban

Breve histórico

- O **kanban**, ou 'sinal visual', surgiu nos anos **1940** na Toyota, no Japão, como parte do **Sistema Toyota de Produção (TPS)**.
 - Originalmente, era um **sistema de cartões físicos** usados para **controlar a produção e o estoque**, baseado na **demanda real**;
 - O conceito foi inspirado nos supermercados, onde itens eram repostos conforme a demanda, **evitando excesso ou falta de estoque**;
 - Um dos principais criadores do TPS e do kanban, Taiichi Ohno enfatizou a importância do **fluxo contínuo e da eliminação de desperdícios**.



Breve histórico

Na prática, de forma simplista, o sistema de cartões físicos era utilizado de forma que:

- Cada cartão representava um trabalho a ser feito;
- Para indicar progresso e status, os cartões eram movidos para outras 'colunas', refletidas no quadro kanban;
 - Dessa forma, ocorria um fluxo contínuo de trabalho, porém levando-se em consideração a demanda e a prioridade, além da capacidade de produção.



Breve histórico

- No início dos anos **2000**, O Kanban foi adaptado para **ambientes ágeis** por David J. Anderson, como uma **abordagem evolutiva para melhorar processos**;
 - Promovendo a ideia de **limitação do trabalho em progresso (WIP) e aprimoramento contínuo**;
- Sua abordagem flexível e adaptável atraiu equipes ágeis por não impor papéis fixos ou cerimônias rígidas;
- Então, equipes que buscavam **flexibilidade e melhoria contínua além das estruturas rígidas do Scrum** começaram a adotar o Kanban;
- E para além do desenvolvimento de software, o Kanban também foi adaptado e adotado em equipes de operações, marketing, recursos humanos e outros setores em ambientes ágeis.

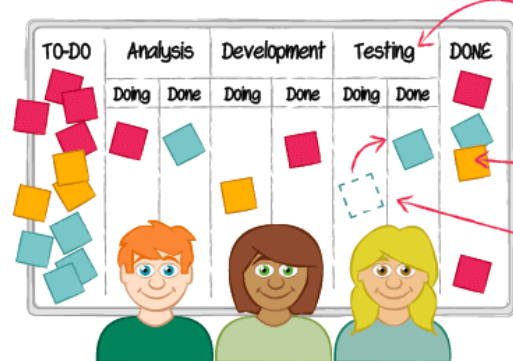


Princípios e como funciona

The 4 principles of

KANBAN

1 Visualize workflow



Label columns to represent either the type of work or who is responsible

Use cards to represent user stories

Move cards from left to right as work gets completed

Visualize your work on a board with **cards** to represent **user stories** in your **product backlog**. Use a colour to represent a **theme**. Place the **cards** into columns depending on their workflow status. As work gets completed, move your **cards** from left to right.

2 Limit work in progress (WIP)



Exceeding your WIP limits will result in a blockage

When you notice a blockage, stop work to assist

Blockages interrupt flow so others can't continue working

Set **WIP limits** to minimise multi-tasking and ensure cards move smoothly across the board. Fill your "TO-DO" column with top priority work from your product backlog, replenishing as necessary. If the flow gets **blocked**, the entire team should stop work, collaborate and fix it.

Key definitions

Card

Red colour represents a theme, e.g web design

Priority

Name of who is performing the work

ID number

Task name

Estimate of effort in hours or other metric

A card should include the information you think represents your user story best.

Theme

Theme	User story
Web design	As an Internet user, I want to use a fast website, so that I can purchase items quickly.
Web design	As an Internet user, I want to use a visually appealing website, so that my overall experience is enjoyable.

A theme is a collection of user stories by category.

User story

As an...	I want to...	So that...
Internet user	use a fast website	I can purchase items quickly

A user story describes a piece of work, speaking from the end user perspective. This serves as a guide to show your team why you are working on something. It encourages everyone to think in terms of business value rather than technicality.

Product backlog

ID	Theme	Story	Criteria	Effort	Priority

A product backlog is an organized version of your to-do list. Each user story includes a set of acceptance criteria (definition of DONE).

Key definitions

Work in progress (WIP)

Work in progress refers to the cards you have started but have not yet finished.



Blocked/blockage

Blockage or bottlenecks can occur if you are unable to finish tasks, causing a queue of incoming tasks and an interruption to the flow.

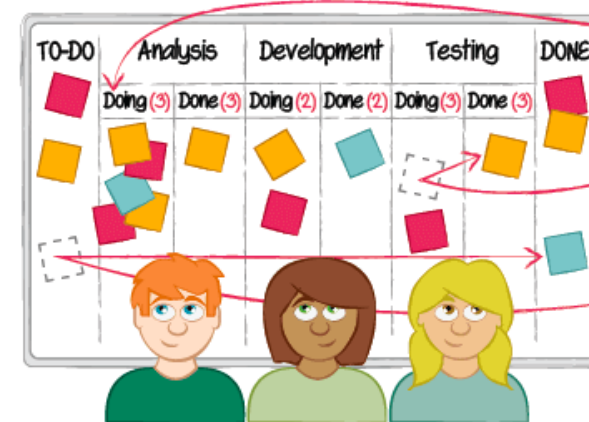


WIP limits

WIP limits (work in progress limits) are the amount of cards that you allow in each column. WIP limits make blockages visible to everyone on the team so that they can be rectified and analysed. It also discourages multitasking and maintains focus.

Doing (3)

3 Focus on flow



Keep a look out for blockages that interrupt workflow

Track cycle time to measure your team's performance

Track lead time to see how long your clients are waiting for delivery

Look out for interruptions in **flow** and use them as opportunities to improve. Your **workflow** should run smoothly, not stop and start. Choose some **flow metrics** to track and analyse them. If your work is flowing smoothly, you know you are creating value.

4 Continuous improvement



Plan how to improve your current method

Implement changes to test them

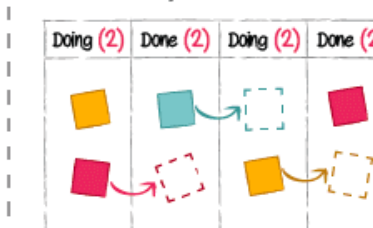
Monitor the results

If successful, implement on a wider scale

Even after implementing Kanban, your work is never truly finished. Part of the Kanban method is to continuously improve your processes. Monitor the way you work and make improvements on an ongoing basis.

Key definitions

Workflow/flow



Workflow describes how your cards are moving across your Kanban board. "Good flow" is when your cards are moving smoothly. "Bad flow" is when movement will stop and start.

Flow metrics

The flow metrics you choose are entirely up to you, but here are some helpful examples:

- Lead time** - how long does it take for a card to move from "TO-DO" to "DONE"?
- Cycle time** - how long does it take for a card to move from "Doing" to "Done"?
- Number of items not started** - are you struggling with your workload?
- Number of items that are WIP** - are you staying within your WIP limits?
- Blockage areas** - do you see any areas where cards build up, causing a blockage in flow?

Get started with Kanban today!

- 1 Visualise workflow
- 2 Limit work in progress
- 3 Focus on flow
- 4 Continuous improvement

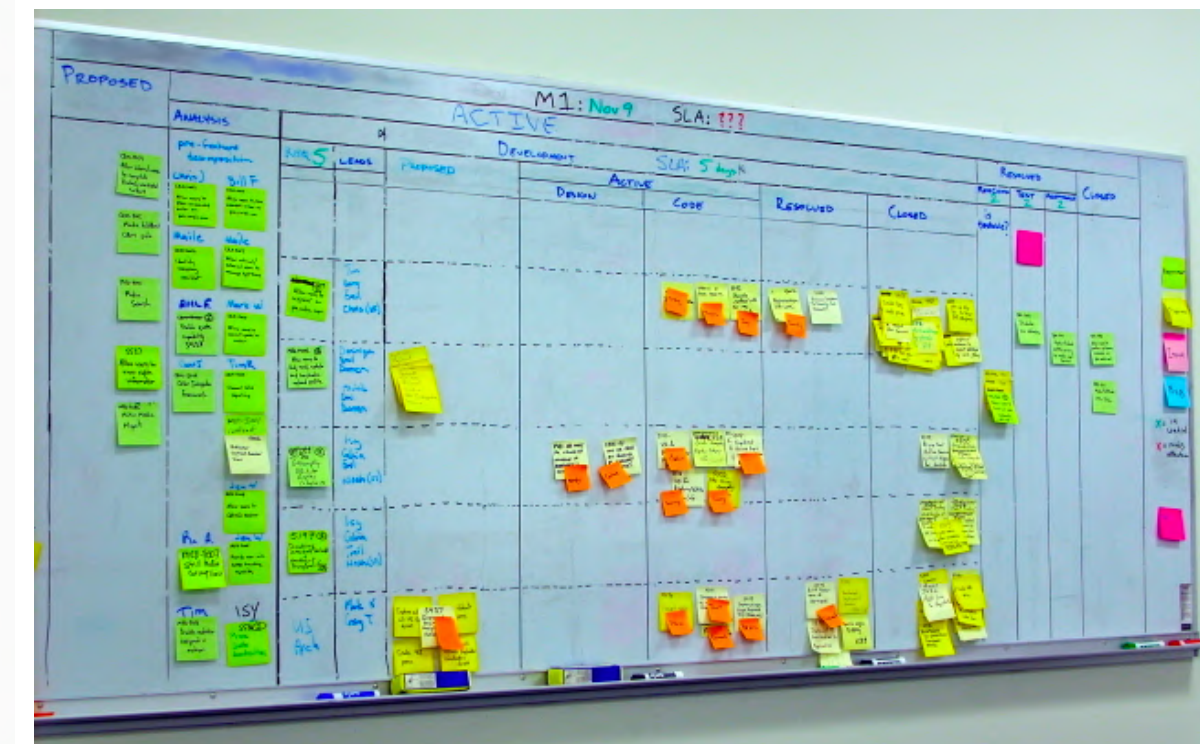
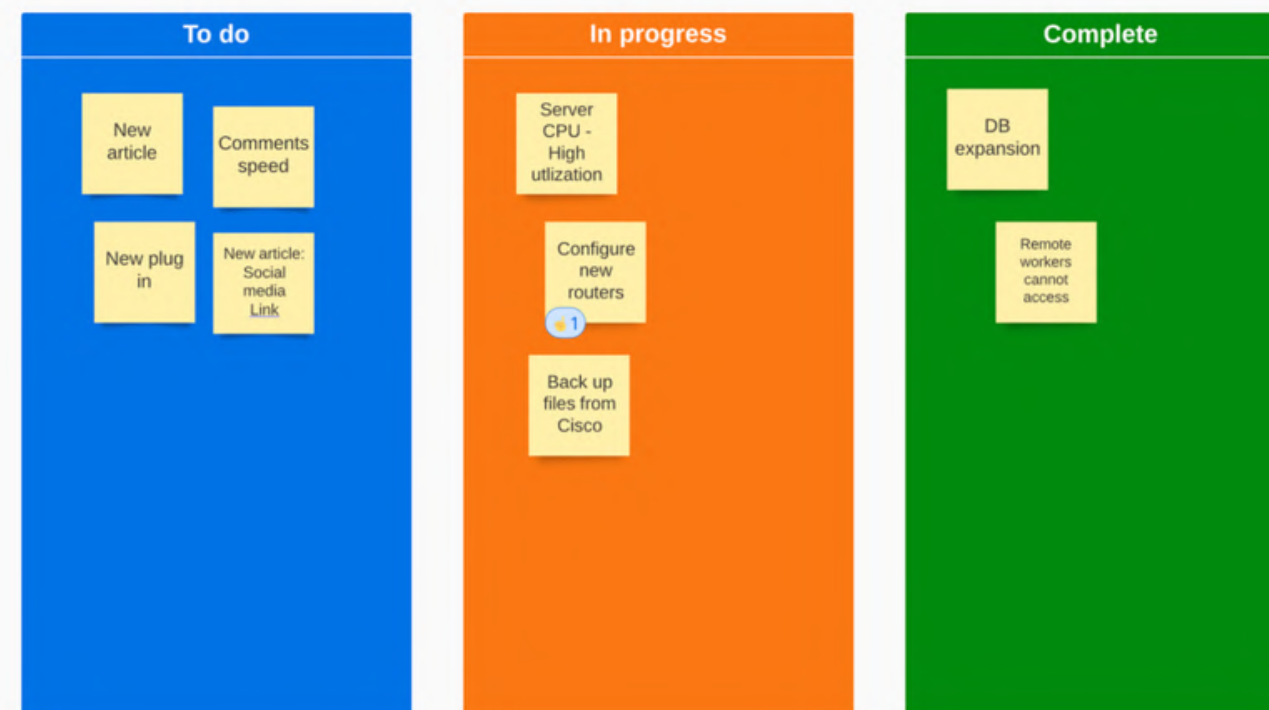
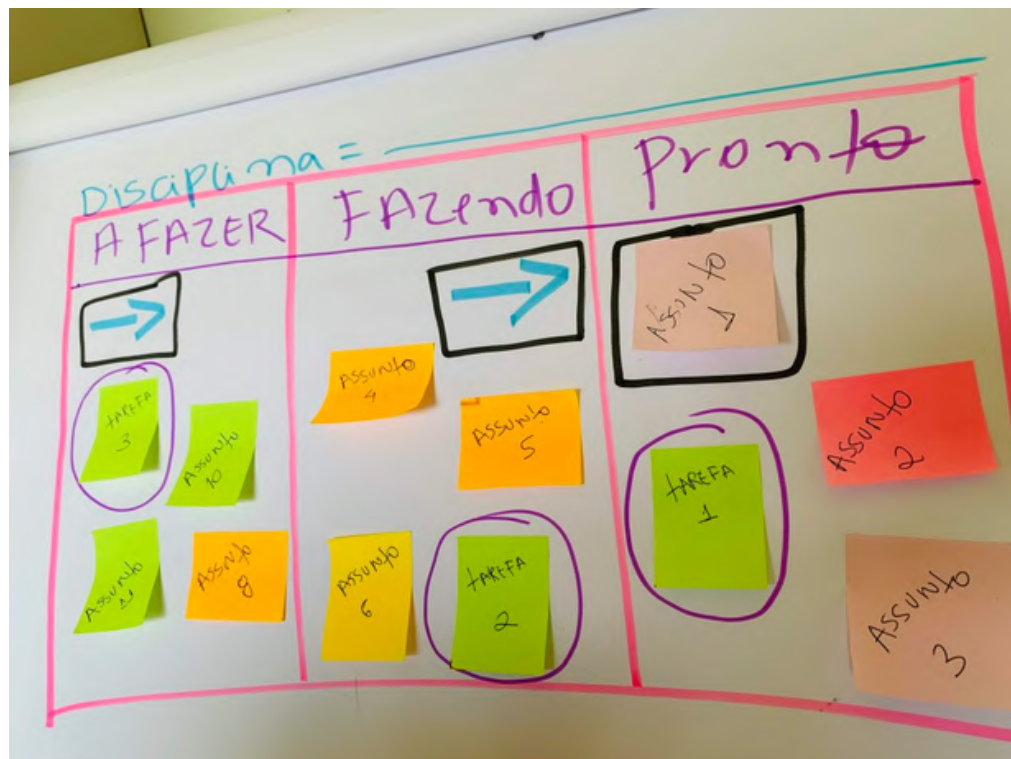
Knowledge
TRAIN

Copyright © 2018 Knowledge Train Limited

Princípios e como funciona

Quadro Kanban

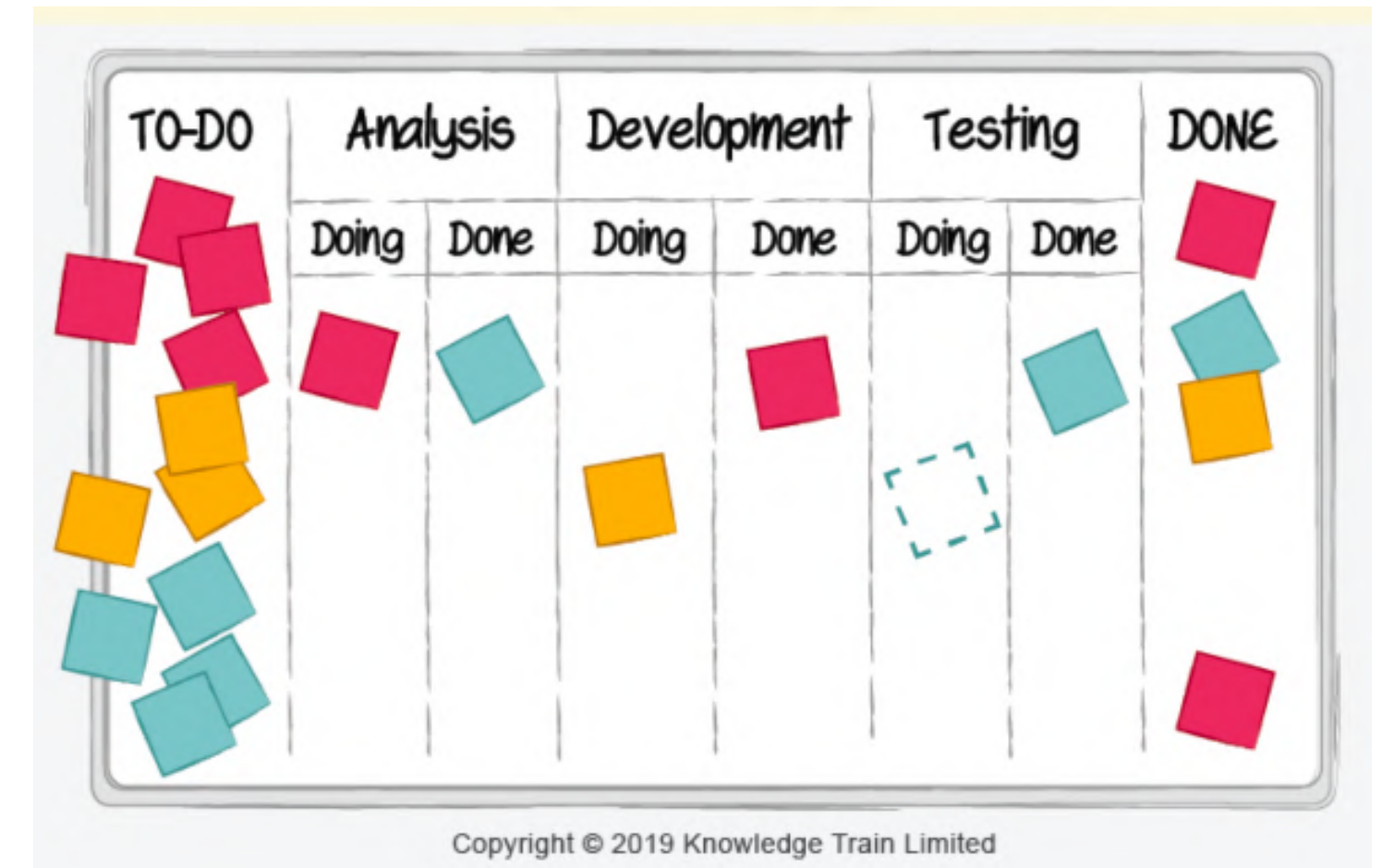
- Um dos princípios do Kanban é a **visualização do fluxo de trabalho por todos os envolvidos**;
- O que, na prática, é a disposição de cards em um quadro, chamado quadro Kanban;
 - Dependendo do contexto, pode ser um quadro físico com post-its, ou um quadro virtual, como por exemplo o <https://kanbanflow.com/>



Princípios e como funciona

Quadro Kanban

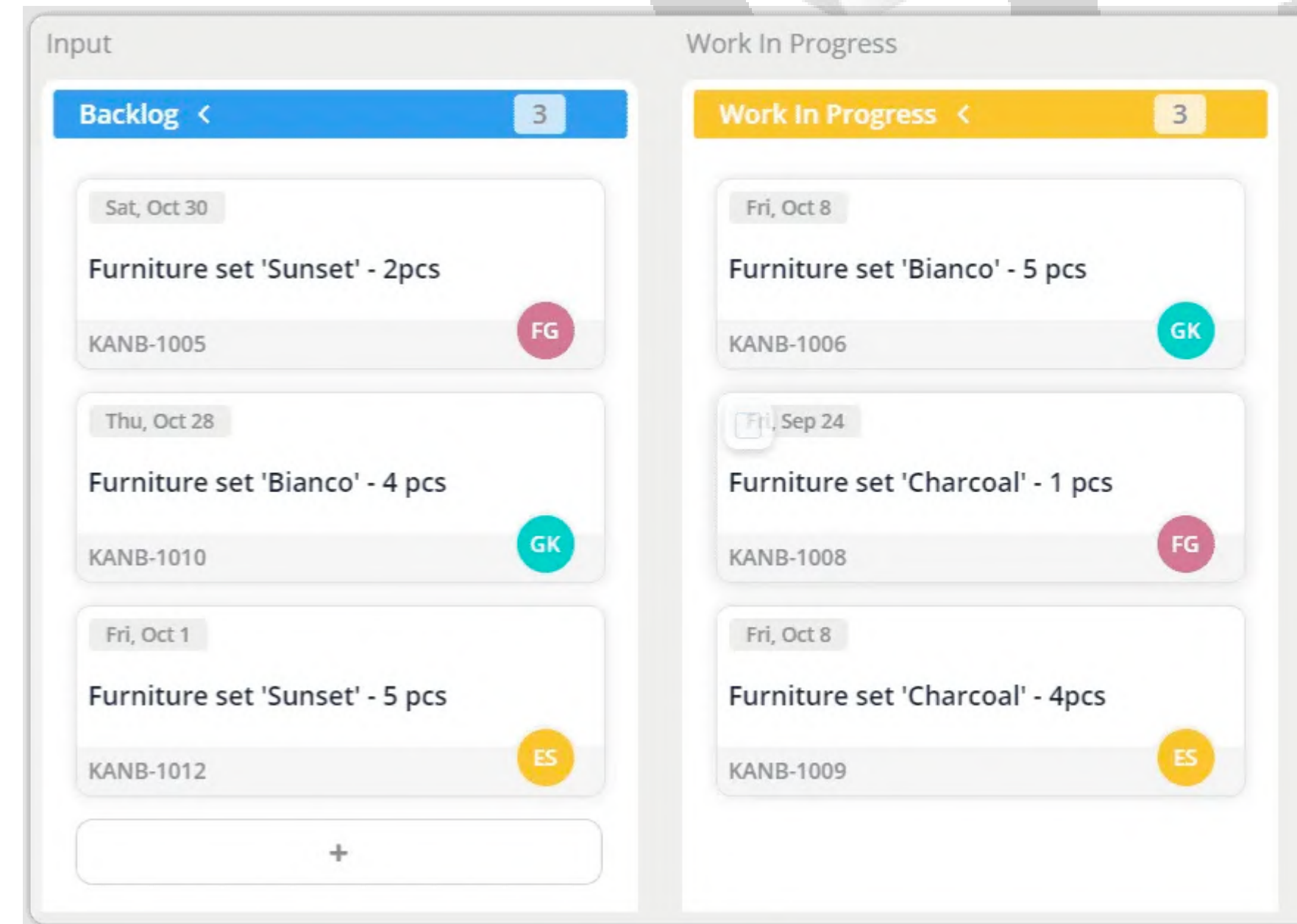
- De forma simplista, **o quadro Kanban é formado por colunas, que representam estágios do desenvolvimento;**
 - Em uma versão mais básica, as colunas seriam: a fazer | fazendo | feito;
 - Contudo, **geralmente mais colunas são adicionadas ao quadro**, de acordo com as necessidades do time;
 - Além disso, também é possível personalizar o quadro para que as colunas sejam agrupadas de forma a demonstrar o time responsável por determinadas etapas, por exemplo.
- **A medida que o trabalho progredir, os cards devem ser movidos entre as colunas**, sempre da esquerda para direita, até a última coluna, geralmente nomeada de "feito", "concluído" ou "done";



Princípios e como funciona

Quadro Kanban

- Quanto aos cards do quadro, **usualmente cada card representa uma user story ou tarefa**;
 - No Kanban, também é possível aplicar o conceito de **product backlog**. E ainda, o próprio backlog do produto pode ser uma coluna a mais no quadro.
- É comum a equipe **estabelecer cores para cada tipo de card**, o que facilita ainda mais a visualização do fluxo de trabalho;
 - Exemplo: pode-se decidir que cards com a cor 'verde' representam melhorias do produto, enquanto cor 'vermelha' representa ajustes mais emergenciais.
- Além disso, **outras informações que um card no quadro pode conter** inclui:
 - O **responsável** pelo card;
 - A **estimativa** de data de entrega, etc.

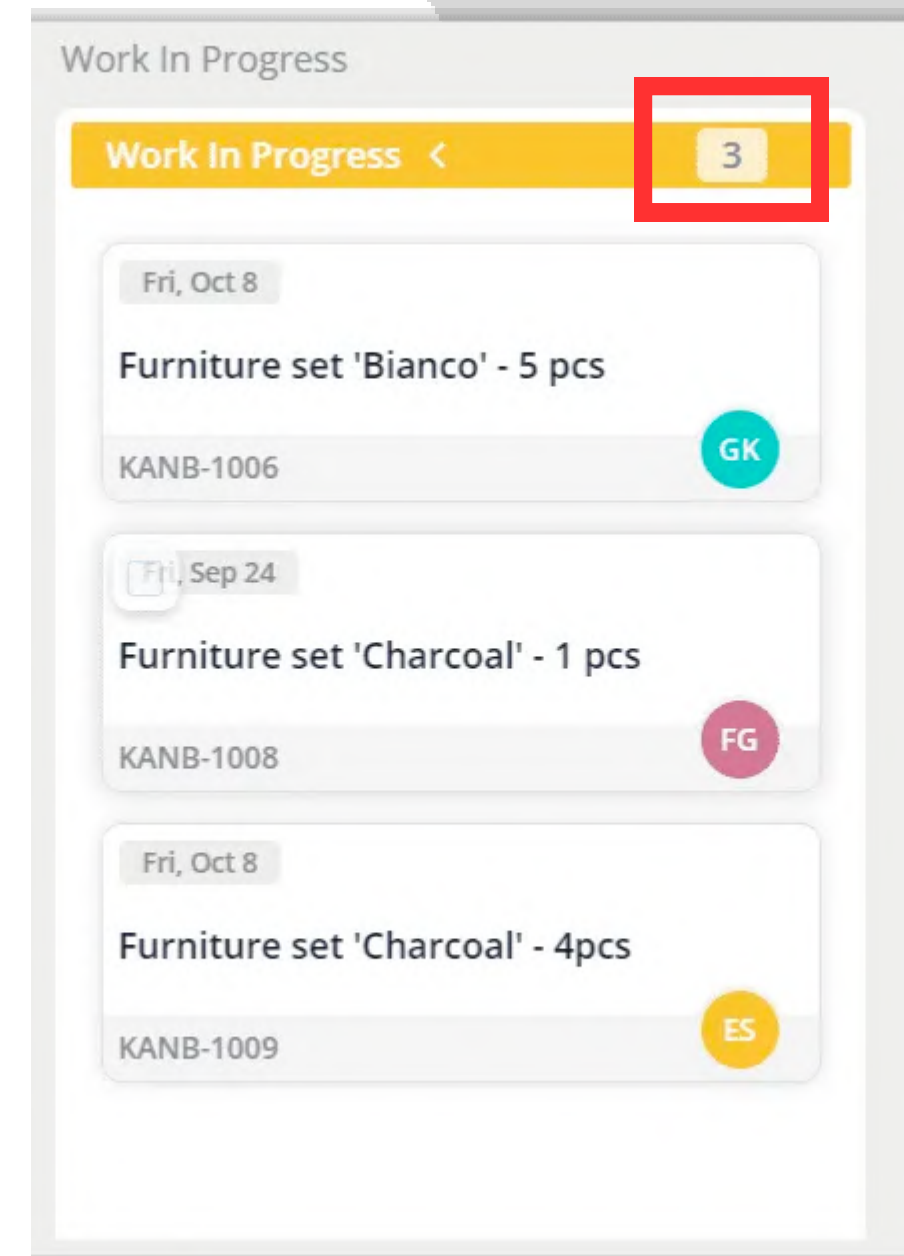


<https://teamhood.com/wp-content/uploads/2021/09/Simple-kanban-board-example.png>

Princípios e como funciona

Limite de Trabalho em Progresso (WIP - Working in Progress)

- Outro princípio do Kanban é **limitar a quantidade de trabalho em status de desenvolvimento**;
- Por funcionar como um **sistema "pull"** (do inglês, puxar), no Kanban uma task só deve ser puxada depois que as tasks em progresso forem concluídas;
 - Para isso, deve-se determinar uma quantidade máxima de atividades por vez;
 - Dessa forma, **a equipe consegue lidar com a demanda considerando a sua capacidade de produção**, o que gera entregas de qualidade, e evita que tarefas fiquem para depois.
- Caso uma task esteja sendo um 'gargalo', **o time deve colaborar para resolver junto da forma mais rápida possível**, para evitar que as demais tasks fiquem bloqueadas.



Princípios e como funciona

Foco no fluxo

- No Kanban, o objetivo é que o **fluxo ocorra de forma contínua e fluida**;
- Portanto, outro princípio é sempre estar com foco no fluxo, olhando para **interrupções como oportunidades de melhoria**;
- Mesmo que não hajam conceitos como 'sprint' no Kanban, ainda é possível realizar a **utilização de métricas**, como por exemplo:
 - **Lead time**: quanto tempo tempo leva para um card mover de 'to-do' para 'done' (tempo de entrega para cliente);
 - **Cycle time**: quanto tempo leva para um card mover de 'doing' para 'done' (performance do time);
 - Número de **itens não iniciados**;
 - Número de **itens em progresso**;
 - entre outras.



Princípios e como funciona

Melhoria Contínua

- Da mesma forma que em outras metodologias ágeis, **a equipe que utiliza Kanban também deve sempre refletir sobre o processo atual para procurar por oportunidades de melhoria;**
 - Mesmo que menos burocrático, o Kanban também possui algumas definições importantes, como o limite de WIP, as colunas do quadro, etc;
 - Portanto, deve-se sempre estar avaliando se o formato atual está funcionando para a equipe, e caso contrário, ajustá-lo.
- No Kanban, não há um ritual específico dedicado a ter esse momento de reflexão, então cabe ao time decidir criá-lo ou deixar que ocorra de forma orgânica a qualquer momento da iteração.



Benefícios e Desafios

- **Benefícios do Kanban:**

- **Visualização do fluxo de trabalho:** Permite uma visão clara e em tempo real das tarefas em andamento, concluídas e pendentes, facilitando a identificação de gargalos e a otimização do fluxo de trabalho.
- **Flexibilidade:** O Kanban é altamente adaptável a diferentes contextos e projetos, permitindo ajustes conforme necessário sem interromper drasticamente o processo.
- **Limitação de trabalho em andamento (WIP):** Ajuda a controlar a quantidade de trabalho que está em progresso ao mesmo tempo, o que pode aumentar a eficiência e reduzir o tempo de conclusão das tarefas.
- **Transparência e colaboração:** Facilita a comunicação entre os membros da equipe, promovendo a colaboração e a responsabilidade coletiva pelas tarefas.

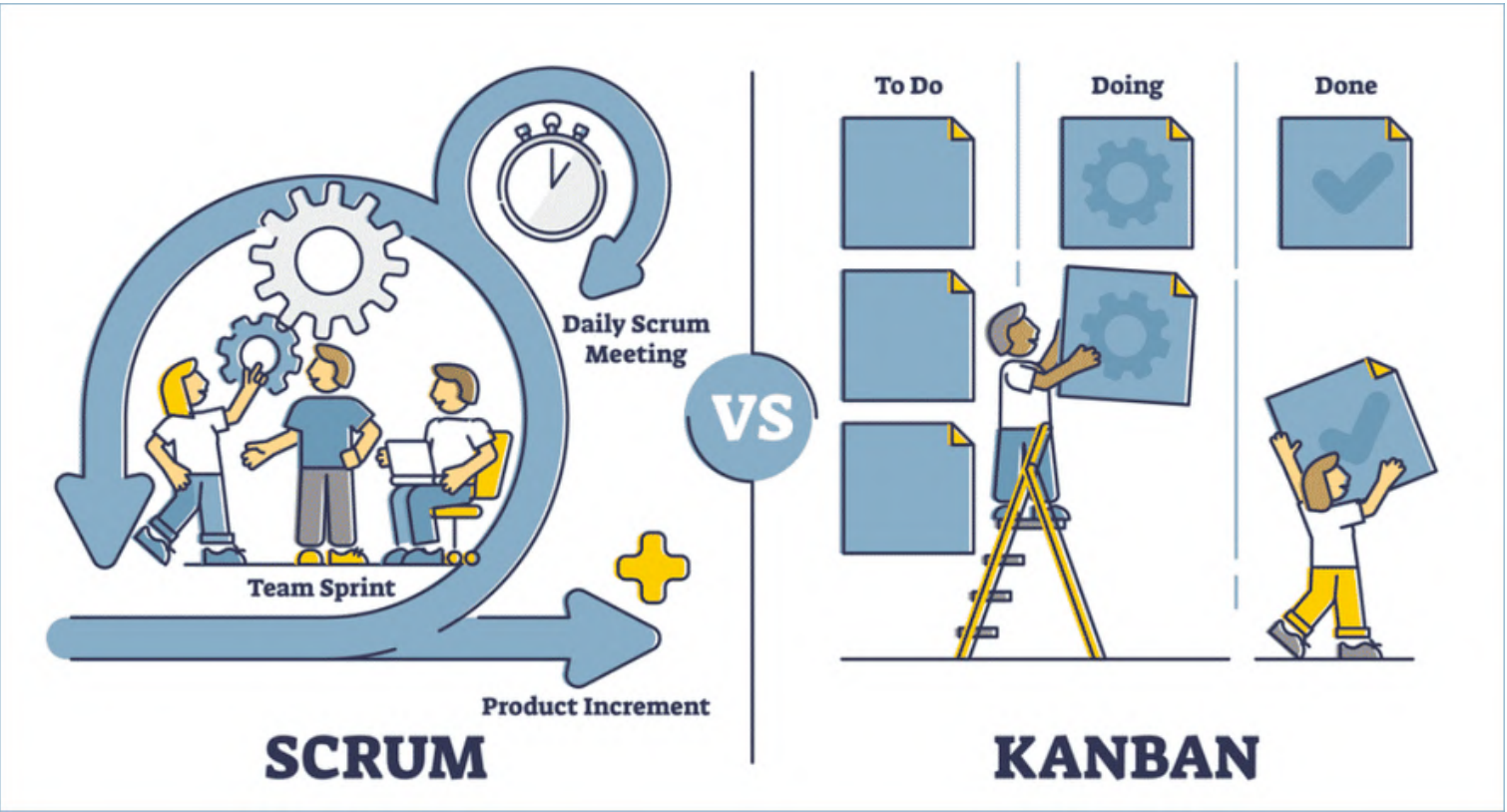


Benefícios e Desafios

- **Desafios do Kanban:**

- **Resistência à mudança:** Implementar o Kanban pode enfrentar resistência por parte da equipe, especialmente se houver uma cultura estabelecida que não valorize a transparência ou a flexibilidade, ou caso a equipe esteja muito habituada a outra metodologia ágil, como o Scrum.
- **Necessidade de atualização constante:** Manter o quadro Kanban atualizado requer comprometimento e disciplina por parte da equipe, caso contrário, pode se tornar desatualizado e ineficaz.
- **Complexidade de processos:** Em projetos complexos ou equipes grandes, pode ser desafiador representar todos os aspectos do fluxo de trabalho de forma simples e eficaz no quadro Kanban.
- **Identificação de prioridades:** Se não houver uma clara identificação e gestão de prioridades, as tarefas importantes podem ser perdidas ou negligenciadas em meio a outras demandas.

Scrum x Kanban



<https://unichrone.com/blog/wp-content/uploads/kanban-vs-scrum.png>

Kanban

Scrum

Funções

Sem funções pré-definidas, mas muitas equipes ainda têm um gerente de projeto

Proprietário do produto, equipe de desenvolvimento, scrum master opcional

Cadência/cronograma

Trabalho flui continuamente de acordo com as necessidades até que o fim do projeto

Depende da duração dos sprints (geralmente em torno de duas semanas)

Metodologia de lançamento

Entrega contínua

Nova funcionalidade/valor entregue ao final de cada sprint

Métricas primárias

Tempo de ciclo, lead time, quantidade de trabalho em andamento

Velocidade e valor criado

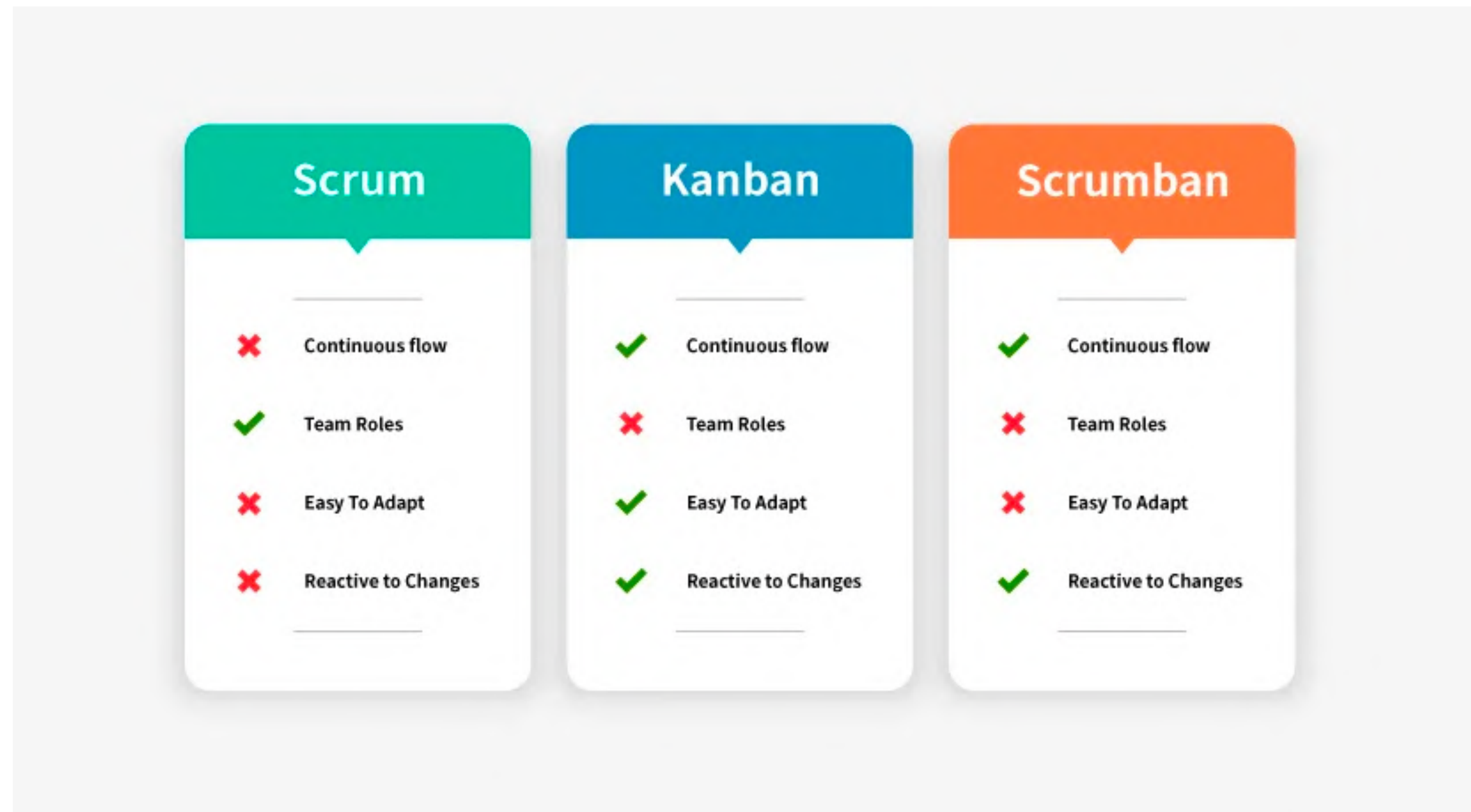
Modificações e alterações

Mudanças podem ser feitas a qualquer momento




Mudanças determinadas e implementadas entre os sprints com base na revisão retroativa. É fortemente desencorajado implementar mudanças no meio de um sprint

Scrumban

- **Scrumban é uma metodologia que combina conceitos do Scrum e do Kanban.**
- Essencialmente, times que utilizam scrumban estão procurando por uma metodologia que seja flexível em relação ao fluxo contínuo e à mudanças;
- Mas que ainda possuam algumas 'regras', como cerimônias específicas do Scrum ou a utilização de planning poker;



Scrumban

	 Scrum	 Kanban	 Scrumban
Easy to adopt	✗	✓	✓
Good for	Big organizations	More experienced Agile teams	Beginners in Agile
Performance metric	Velocity	Cycle time, lead time	Cycle time, lead time
New roles & events	✓	✗	✗
Daily progress tracked	✓	✗	✗
Estimations	✓	✗	✗
Prioritization	During backlog refinement	Optional	During on-demand planning
Any task size accepted	✗	✓	✓
WIP limits	✗	✓	✓
Visual board	✗	✓	✓
Pull system	✗	✓	✓
Release cycle time	Slower	Optimized	Optimized

Desenvolvedor(a) Jr | Remoto

Plugaaê · Florianópolis, Santa Catarina, Brazil (Remote)

Apply

Save

Vaga para Desenvolvedor Júnior no setor de Integrações da empresa,

O Que Você Vai Fazer

- Entendimento e documentação de requisitos de software;
- Criação, documentação e execução de processos do setor;
- Documentação dos serviços de integração com os quais a empresa trabalha;
- Prestar suporte a integradores externos;
- Planejamento e gerenciamento de versões utilizando a metodologia Scrum;
- Desenvolvimento de novas integrações Implementação de melhorias e

Junior Software Engineer

Trimble Inc. · Brazil (Remote)

Apply

Requisitos

- Conhecimento em Java
 - Conhecimento em microsserviços
 - Conhecimento em JavaScript / TypeScript
 - Conhecimento em Testes Unitários
 - Conhecimento em metodologia ágil de desenvolvimento
 - Conhecimento em soluções na nuvem (AWS, Azure ou Google Cloud)
 - Conhecime
 - Inglês avar
- Pessoa Desenvolvedora J...

Luizalabs · Brazil (Remote)
- Easy Apply

Save

- Tecnologias: .NET Core, JavaScript (ES6), HTML5, CSS3
- Bancos de dados SQL
- Controle de versão com Git (Gitlab, Github, Gitflow)
- Arquiteturas baseadas em APIs e microsserviços
- Paradigmas de programação, padrões de projeto e boas práticas
- Técnicas como TDD, BDD, Clean Code, SOLID

Diferenciais

- Testes automatizados (unitário, integração, E2E, stress)
- Automação, Integração contínua (CI) e Deploy contínuo (CD)
- Bancos de dados NoSQL (MongoDB), Redis
- React (controle de estado, hooks, context api...);
- Metodologias ágeis;
- Containers com Docker e Kubernetes

Programador Cobol Junior

Unisys · Brazil (Remote)

Apply

Save

O que esperamos de você:

- Graduação completa na área de Tecnologia
- Conhecimento e experiencia com desenvolvimento de aplicação utilizando Cobol
- Conhecimento e experiencia de trabalho em equipe com métodos ágeis SCRUM ou Kanban
- Experiência com processos da engenharia de software e desenvolvimento c sistemas;

Desejável ter:

- Experiencia desenvolvimento ambiente mainframe Unisys, COMS, DMSII, SI CANDE
 - Conhecimento em testes unitários e de integração automatizados (TDD/ATDD) em rotinas COBOL.
 - Ser flexível, colaborativo e proativo;
 - Empatia e capacidade de colaborar em equipe;
 - Buscar apr
 - Iniciativa, t
- Junior Software Engineer in Te...

Trimble Inc. · Brazil (Remote)
- Apply

Save

Requisitos

- Conhecimento de metodologias Ageis;
- Conhecimento em ferramentas de gerenciamento de casos de teste;
- Conhecimento em automação de testes;
- Conhecimento em testes não funcionais (performance, segurança, etc.);
- Conhecimento em uma linguagem de programação (Java, Kotlin, JavaScript/TypeScript,Pyrhon ou outras);
- Conhecimento em soluções na nuvem (AWS, Azure ou Google Cloud);
- Conhecimento em teste de aplicações Web;
- Conhecimento em testes de backend;
- Inglês avançado

Pessoa desenvolvedora Frontend - Jr.

Petlove · Brazil (Remote)

App

Vivência nas seguintes tecnologias:

- VueJS (ou algum outro framework de front-end. React)
- GIT
- Javascript (Vanilla)
- HTML 5
- CSS 3 / Scss
- Testes unitários

Pessoa Desenvolvedora de Software Júnior...

Grupo OLX · São Paulo, São Paulo, Brazil (Hybrid)

Apply

Save

Hoje contamos com dois escritórios nas cidades de Rio de Janeiro e São Pa o #TáPorOnde?, nesse modelo de trabalho, nossas pessoas podem trabalha qualquer lugar do Brasil.

Responsibilities and assignments

Você fará parte do time que é responsável por cuidar da jornada de marketp

No dia a dia você vai:

- Construir e melhorar features dos portais/aplicativo
- Olhar para implementação de novas regras de negócio
- Realizar ajustes de layout dos portais/aplicativo
- Realizar o deploy das aplicações

Requirements And Qualifications

E o que você precisa ter?

- Conhecimentos teóricos de boas práticas e processos de engenhari principalmente, testes, versionamento;
- Conhecimentos básicos em engenharia de software;
- Conhecimento sobre versionamento de código usando Git;
- Conhecimentos básicos sobre o funcionamento da plataforma web;
- Conhecimentos básicos em JavaScript, HTML e CSS
- Conhecimentos básicos sobre o funcionamento da plataforma mobil
- Conhecimentos em React Native.

Pode ser um diferencial/desejável:

- Conhecimento sobre integração contínua;
- Vivência com uma aplicação desde o desenvolvimento e produção;
- Conhecimento sobre metodologias ageis e seus rituais;
- Conhecimentos basicos sobre arquitetura de software;

Quadro Kanban: Prática 03

Vamos praticar?



1. **Organizar um novo projeto com template Kanban no Jira;**
 - a. Note que, por padrão, não há um espaço de 'backlog'.
2. **Mover todas as user stories e/ou tarefas da prática 02 para a lista de 'to-do' do quadro Kanban, de forma organizada e priorizada.**
 - a. Caso a equipe tenha feito o refinamento de mais de 6 user stories, nesse caso deve-se mover apenas 6 (seis).
 - b. Os cartões devem estar em ordem de prioridade, sendo que o cartão mais ao topo possui a prioridade mais alta;
 - c. Para realizar essa organização, deve-se:
 - i. Discutir em grupo sobre o escopo do projeto como um todo;
 - ii. Levar em consideração quais tasks são bloqueantes de outras;
 - iii. Considerar o limite de WIP (tópico 03 abaixo).
 - d. Deve-se anotar as discussões entre o grupo e justificar o racional por trás da priorização das tasks como um todo.
3. **No quadro Kanban, na coluna doing, deve-se definir um limite de trabalho em progresso (WIP)**
 - a. O grupo deve discutir as habilidades dos membros que fazem parte, a quantidade de tasks e o escopo do projeto, e justificar o número decidido para ser o limite de WIP;
 - b. E, no Jira, alterar na coluna em si a quantidade máxima de itens para o número decidido.

Obrigada!

Dúvidas?

- **Slack**
- **Email:** jlslc@icomp.ufam.edu.br



Vamos manter contato =)

<https://www.linkedin.com/in/julialuiza/>