



Curso

Qualidade de software



Qualidade de Software

Módulo 7



Eventos ágeis, DoR e DoD

Aula 1

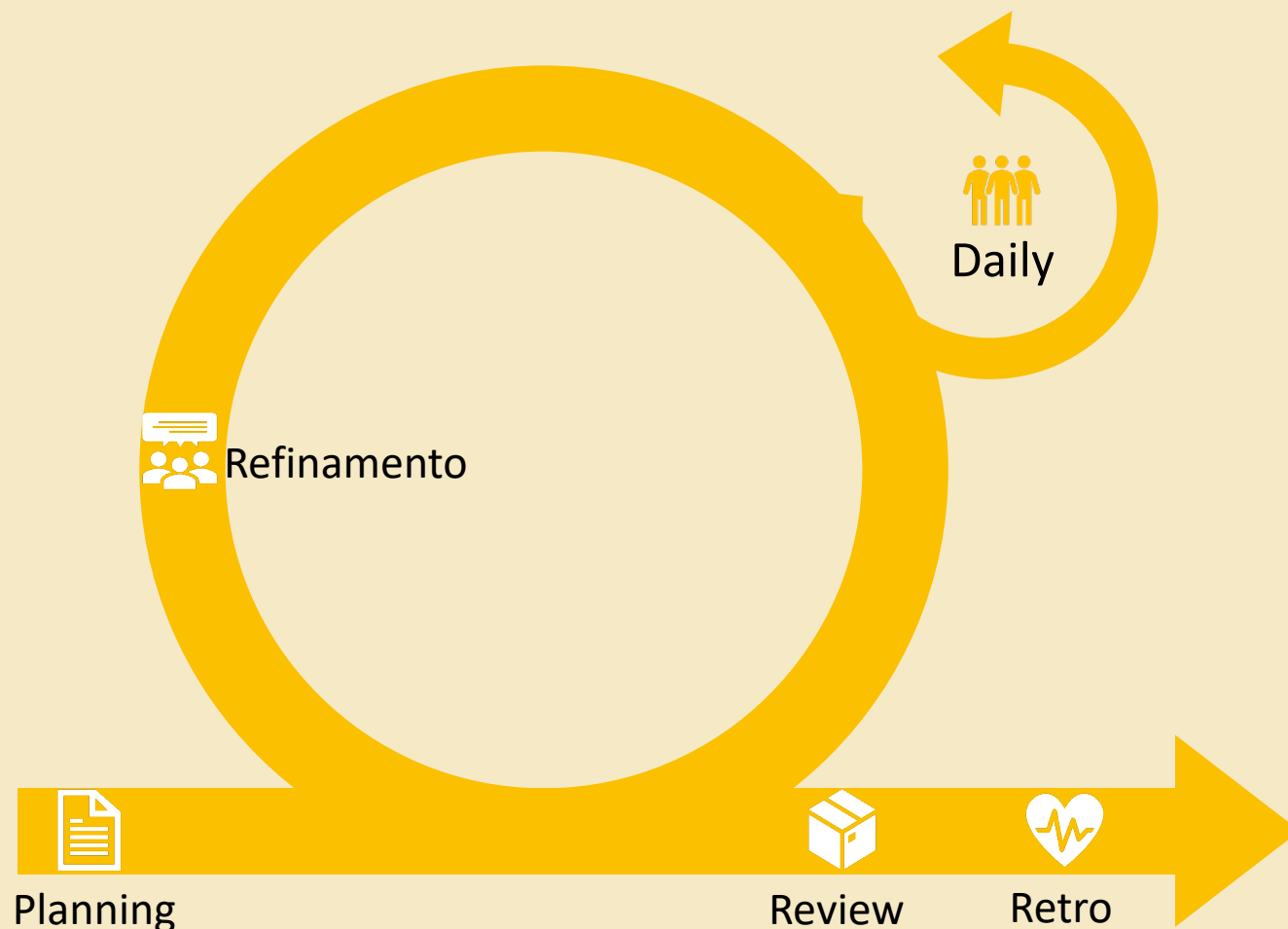
Eventos do Scrum

Principais:

- Planejamento de Sprint (planning)
- Reunião diária (Daily)
- Revisão do Sprint (Review)
- Retrospectiva do Sprint (retrospective)

Opcionais, porem super valiosos:

- Refinamento (Refinement)
- Três amigos (do BDD)

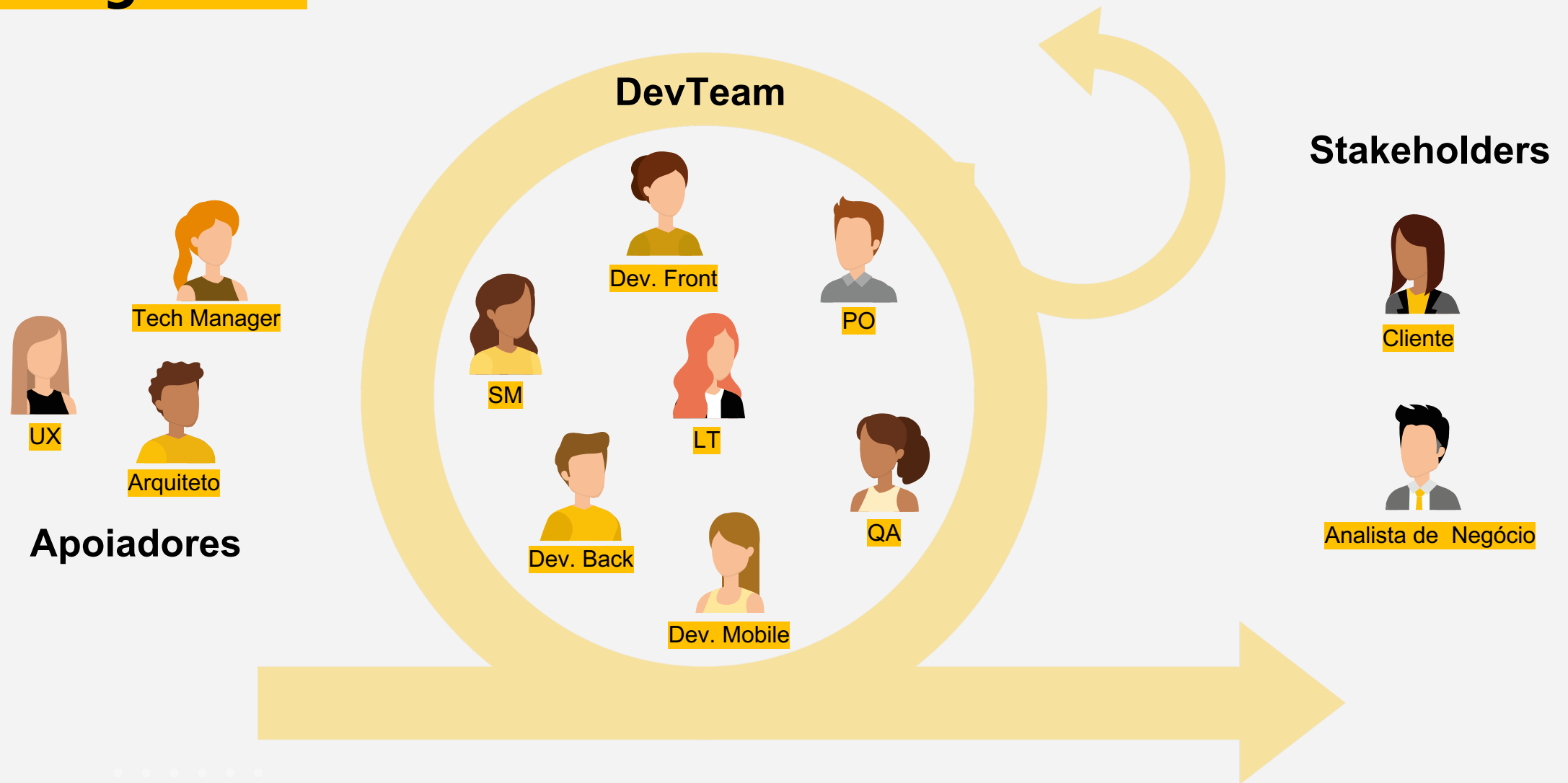


Objetivos dos eventos

- Padronizar os eventos e manter uma cadência;
- Diminuir a necessidade de reuniões, pois esses eventos já preveem o momento certo para a troca de informações necessárias
- Feedbacks contínuos entre os participantes;
- Aumentar a colaboração entre os membros do time;
- Transparência dos status das atividade.



Integrantes dos eventos



Início do projeto

- **SM** junta o time e apresenta a metodologia de trabalho, assim como as práticas;
- **PO**, apresenta o projeto (jornada do produto), desafios e o objetivo do produto;
- **Desenvolvedores** apresentam suas expertises;
- **QA**, fala de sua experiência, qual sua expectativa com o time, com o projeto e como ele pode ajudar o time a atingir a excelência;



SM



Dev. Front



PO



QA



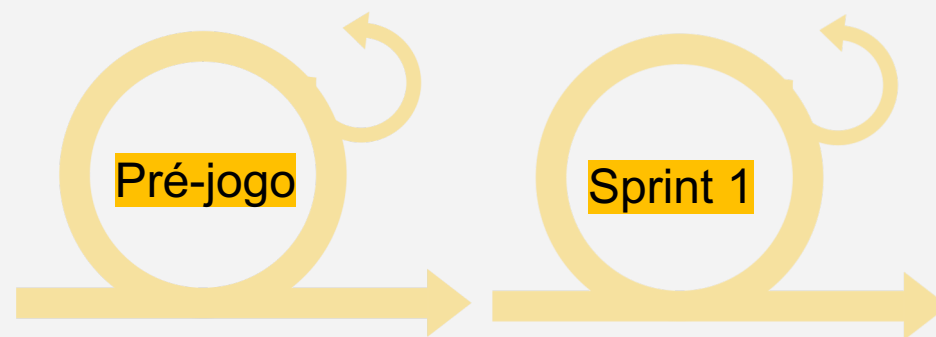
Dev. Back



Pré-jogo

Objetivo:

- Ajustar os horários do time, dinâmicas de eventos e interação entre os membros do time;
- Treinamentos de ferramentas, processos, tecnologias e/ou ambientes internos;
- Desenvolvedores apresentam arquitetura do projeto (tecnologias e linguagens de programação, frameworks) que irão usar;
- O QA defini o projeto de automação de testes, artefatos e repositório dos plano de testes e apresenta para o time;
- Desenvolver um pequeno módulo para validar alguns itens;
- Definir os critérios de entrada e saída (DoR e DoD);



Definition of **Ready**

Todas as premissas necessárias para que o time comece o desenvolvimento (preparada):

Exemplo de DoR:

- Histórias de interface gráfica com Protótipo desenhado e anexado;
- Histórias de API de parceiros, com a URL da documentação;
- Todos os itens com **critérios de aceite**;
- Histórias devem ser **INVEST**;
- Histórias tem que ter sido refinada;
- Todos os itens acordados com a área de negócio.



Definition of **Done**

Todas as premissas necessárias para a história ser considerada como entregue (pronta)

Exemplo de DoD:

- Cobertura de testes de unidade mínima de 80%;
- Testes automatizados dos **critérios de aceitação**;
- Todos os cenários de testes (funcionais e não funcionais) executados e aprovados;
- Funcionalidade documentada na Wiki;
- Histórias de APIs devem estar com Health Check;
- Funcionalidade publicada em Produção;





Planning e Story Points

Aula 2

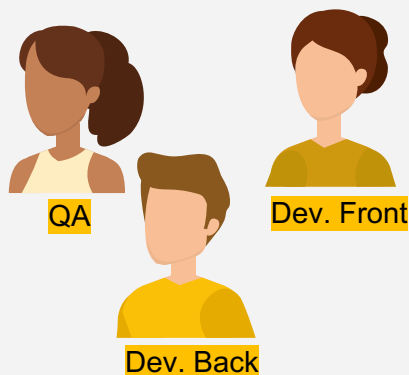
Reunião de planejamento - **Planning**

Serve para estimar e decidir todo o trabalho que será desempenhado pelo DevTeam durante a Sprint.

Facilitador: **Product Owner**



DevTeam



Papel do **DevTeam** neste evento:

- Entender o objetivo principal da Sprint;
- Auxiliar o PO a identificar possíveis problemas;
- Estimar as histórias de uma forma coerente e possível de ser entregue.

Perguntas frequentes

- Podemos levantar dúvidas que não foram retiradas no refinamento?

☒ SIM

☐ NÃO

- Podemos recusar uma história de usuário?

☒ SIM

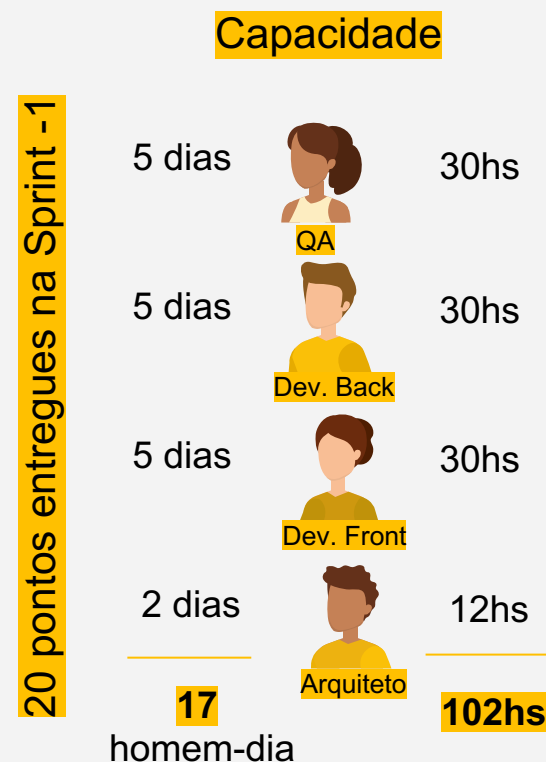
☐ NÃO

- Como saber quantas histórias de usuário cabem em uma Sprint?

R.: Calculando a capacidade e velocidade do time

- Eu QA, devo pontuar as atividades junto ou separado dos desenvolvedores?

R.: Juntos!!!



Planning Poker

Cartas de Planning Poker



0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55...

$2 + 3 = 5$



Cartas do planning Poker



0

1/2

Tão simples que não irá demandar tempo suficiente para atrapalhar a entrega

1

2

3

5

8

13

20

Indica a complexidade e esforço das tarefas

100

∞

Tão grande e complexa que não tem como ser estimada.

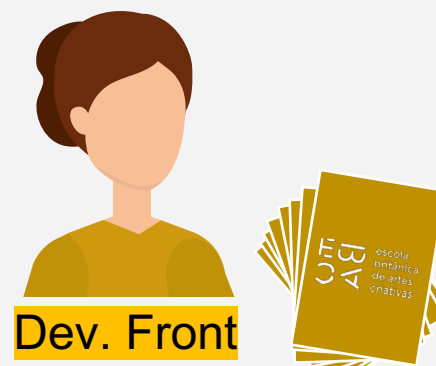
?

Quando se tem dúvida do que foi determinado na História de usuário



Aviso de que o time está cansado, sugerindo continuar mais tarde.

Como jogar?



Como jogar?



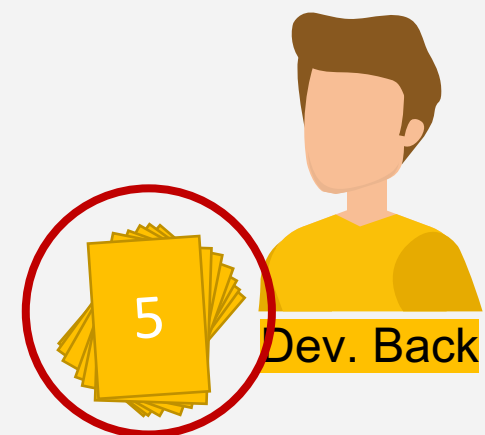
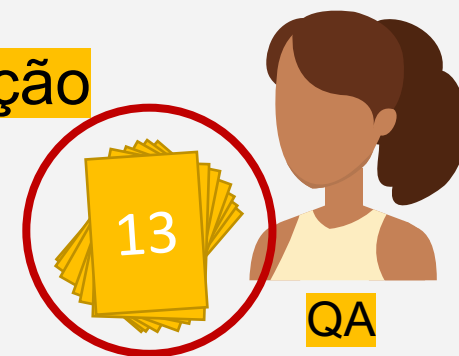
Estimativa pronta



Como jogar?



Divergência na pontuação



Como jogar?



Estimativa pronta

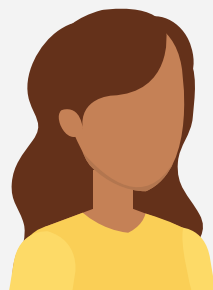


Como jogar?

E quando não há consenso?



PO



SM



Dev. Front



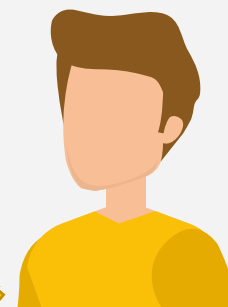
8



QA



13



Dev. Back



5

Promove um acordo

T-Shirt Size

- É uma outra técnica para medir o esforço das Histórias de usuário;
- Não usa número para estimar;
- Dessa forma o time fica mais livre pra pensar de forma abstrata;
- Aqui as tarefas são organizadas como tamanho de camisetas;
- O time decide qual tamanho usar para cada História;



Como usar?

- Primeiramente identifique a unidade de medida em horas;
- A unidade de tempo vai depender da velocidade do time;
- Cada time pega os cartas com as opções.



Cartas de T-Shirt Size



Ex. Até 2hs

De 2 a 4hs

De 4 a 8hs



Tarefas pequenas



Tarefas muito grandes

Como **usar?**

- Depois o time pega um História de referência, de preferência uma pequena;
- E vai estimando o tamanho das outras histórias.



Pós planning para o QA

- Divisão das tarefas (subtasks);
- Fazer o planejamento dos testes (word, planilha ou mind map);
- Criar cenários de testes, baseado nos critérios de aceitação;
- Levantamento de massa de dados;
- Divisão dos testes automatizados (unit, integration, ui);

Tudo isto num período após a planning.





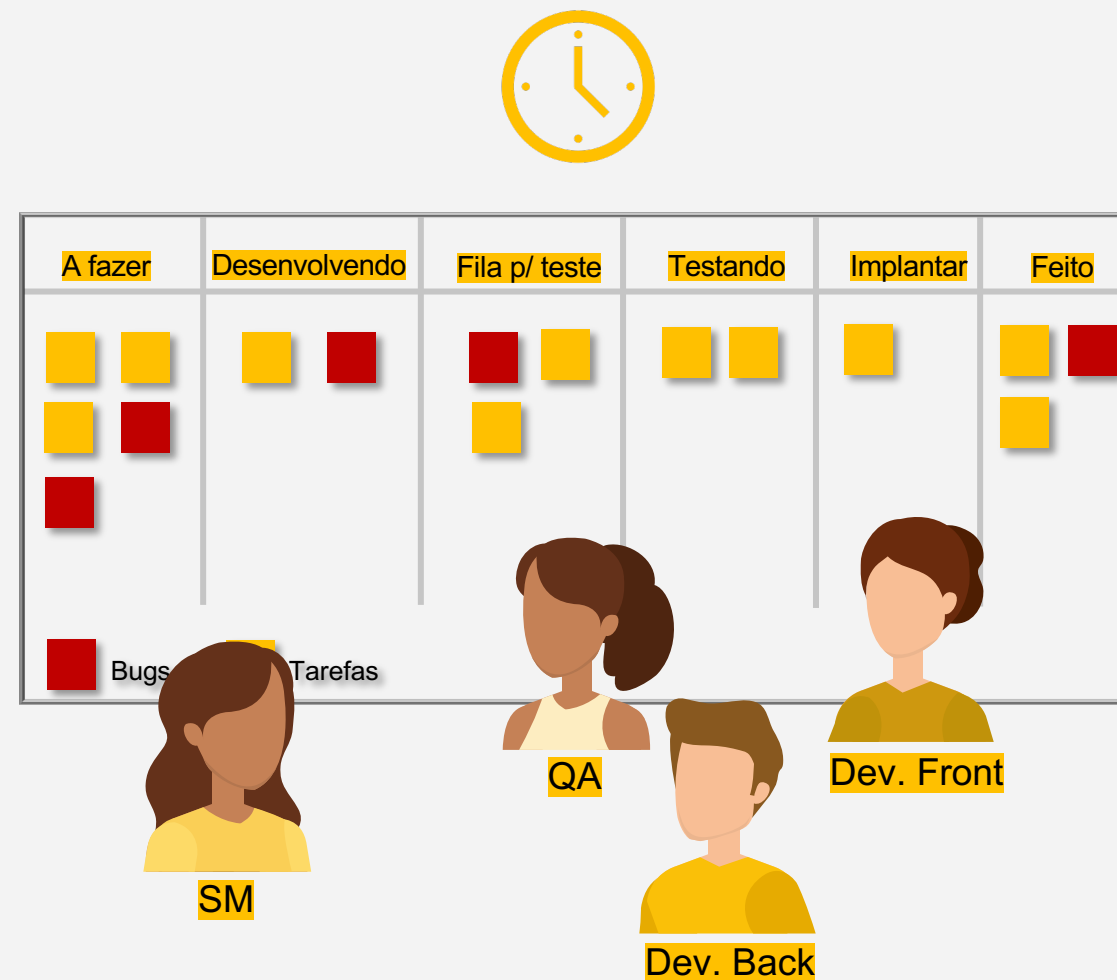
Daily e Burndown

Aula 3

Daily – Reunião diária

Facilitador: DevTeam

- Objetivo: Time reforçar o planejamento para as próximas 24 horas, com o foco em atingir a meta da Sprint;
- Quando presencial, recomendável fazer de pé, em frente ao quadro de atividades (kanban);
- O horário é decidido pelo time, idealmente o mesmo horário todos os dias;



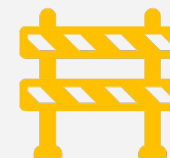
Sugestão de perguntas



O que fiz ontem que ajudou a equipe de desenvolvimento a atender a meta da sprint?



O que eu farei hoje para ajudar a equipe de desenvolvimento a atender a meta da sprint?



Eu vejo algum obstáculo que impeça o desenvolvimento no atendimento da meta da sprint?



Acompanhamento das atividades

- Sem métricas o time simplesmente pode rodar o ágil na sua forma mais básica, mas não estará melhorando continuamente o que é um **grande risco**;
- Uma forma simples de visualizar o andamento das atividades é através do **Gráfico Burndown**;
- Os gráficos burndown são usados para prever a probabilidade da sua equipe de concluir o trabalho no tempo disponível;



Gráfico Burndown

- 1. Estatística de estimativa:** o eixo vertical representa a estatística de estimativa selecionada;
- 2. Valores remanescentes:** a linha vermelha representa a quantidade total de trabalho deixado no sprint, de acordo com as estimativas da sua equipe;
- 3. Orientação:** a linha cinza mostra uma aproximação de onde a equipe deve estar, assumindo um progresso linear.

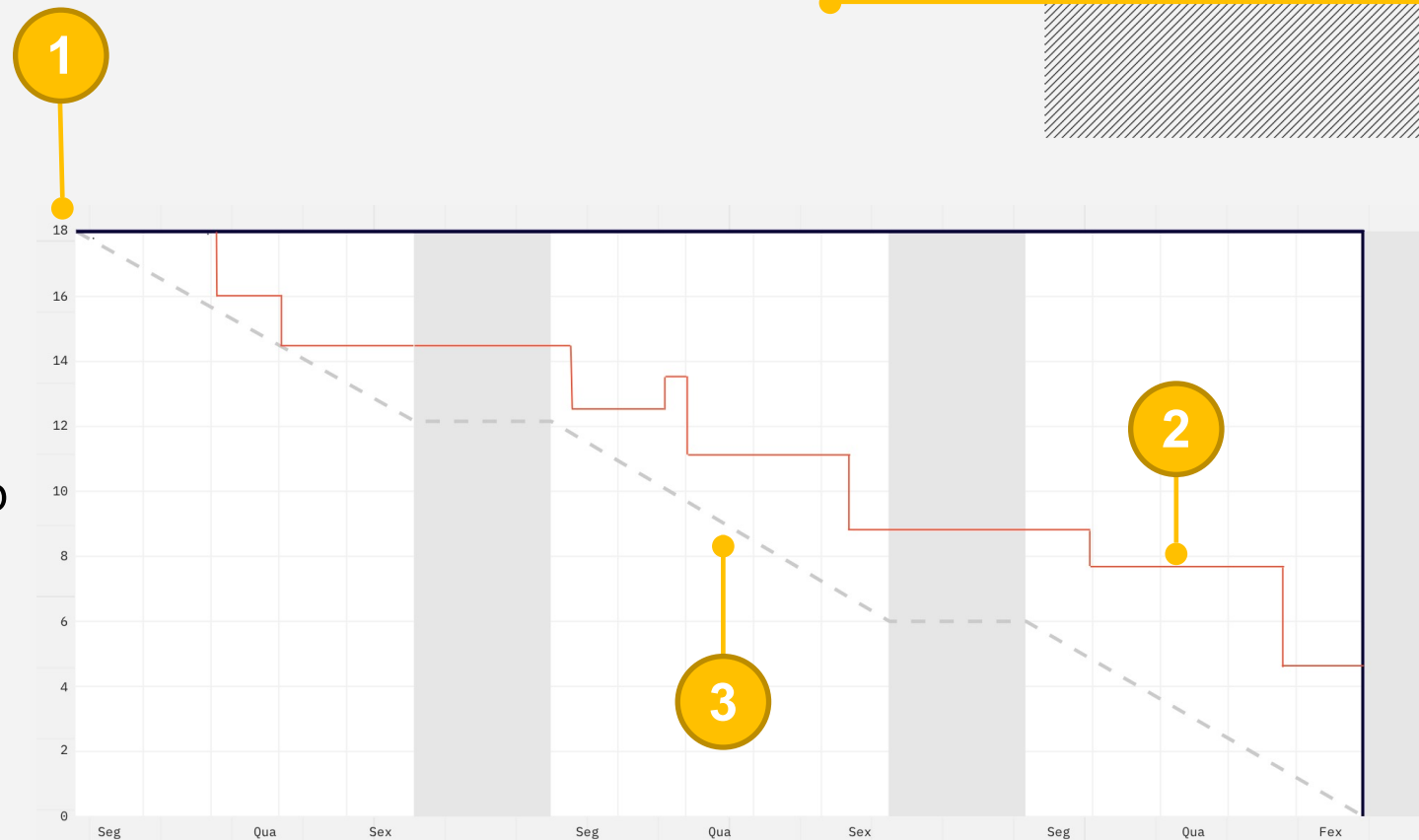
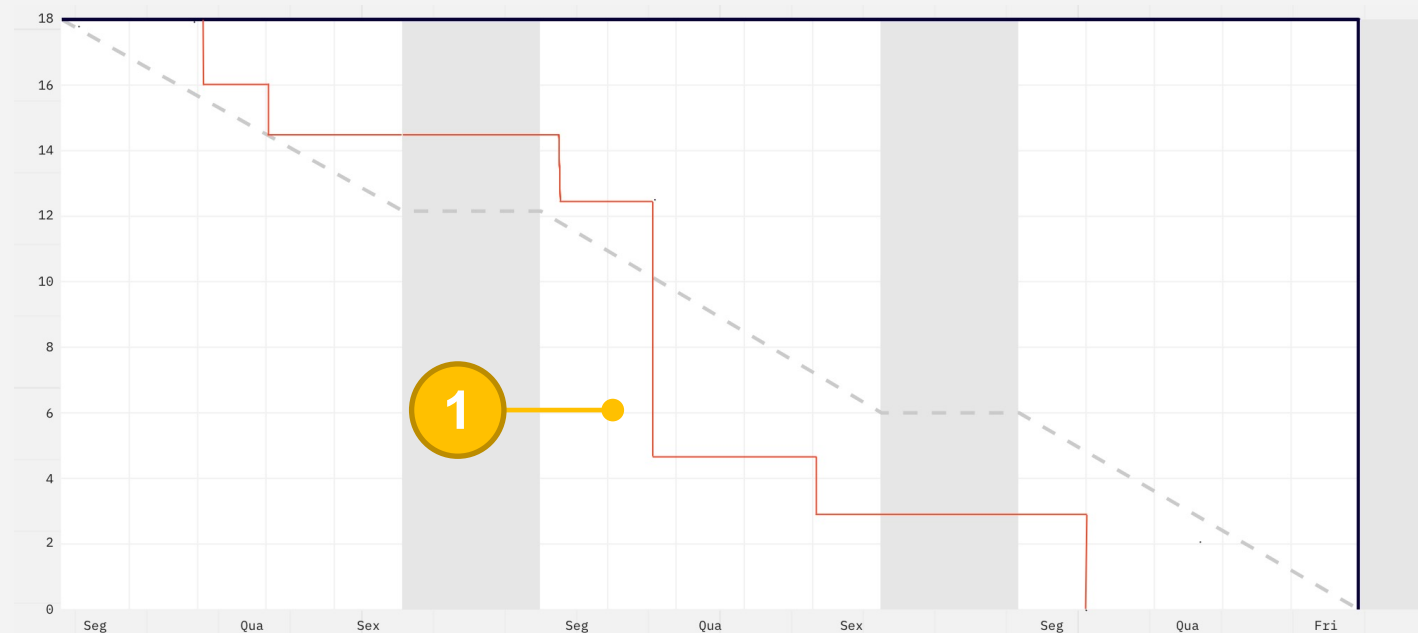


Gráfico Burndown

1. Se a linha vermelha estiver abaixo dessa linha, indica que o time está no caminho para concluir todo o trabalho até o final do Sprint;
2. Mas é preciso acompanhar para continuar no ritmo





Review e Retrospective

Aula 4

Sprint Review – Revisão

- Objetivo principal é o time apresentar e avaliar o trabalho que foi desenvolvido durante a Sprint;
- Realizado ao fim da Sprint;
- DevTeam deve responder as dúvidas do PO;
- Provar ao PO que todos os **critérios de aceite** de uma historia foram concluídos com sucesso;
- DevTeam + PO + SM devem comparecer, mas também pode ser aberto às demais pessoas interessadas (Stakeholders);



De que forma?



O time demonstra história a história, passando pelos critérios de aceitação e funcionalidades desenvolvidas.



Deve ser demonstrado num ambiente mais próximo ao de produção



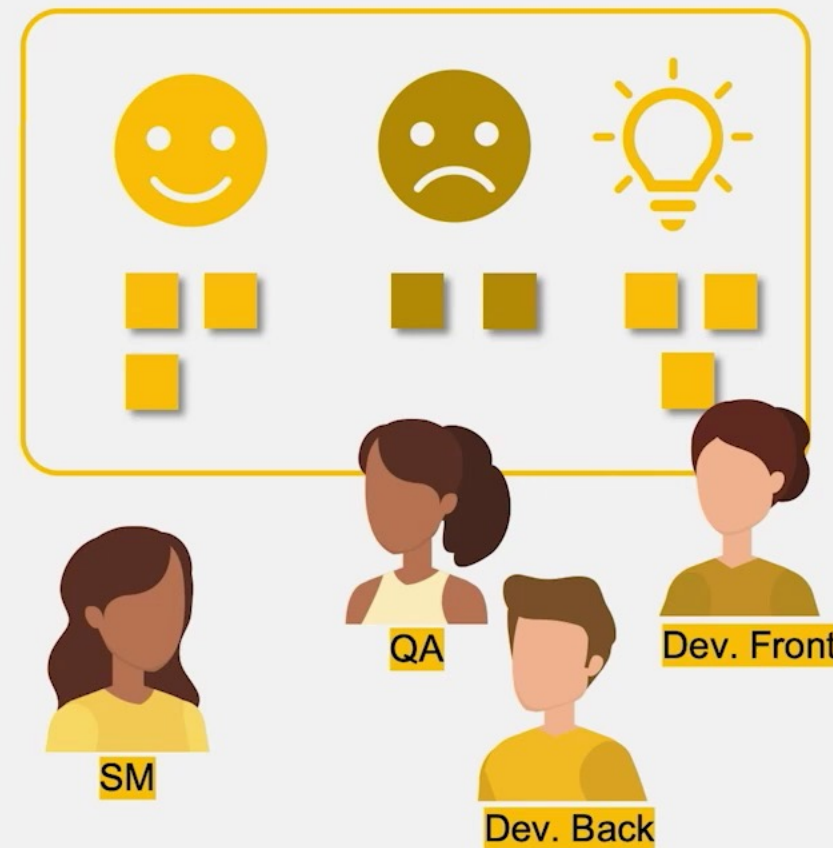
Pode ser uma apresentação em slides ou gravação da tela e pode ser apresentado por qualquer um do time



O PO pode não aceitar o desenvolvimento. Neste caso, deve-se repontuar a história na próxima sprint

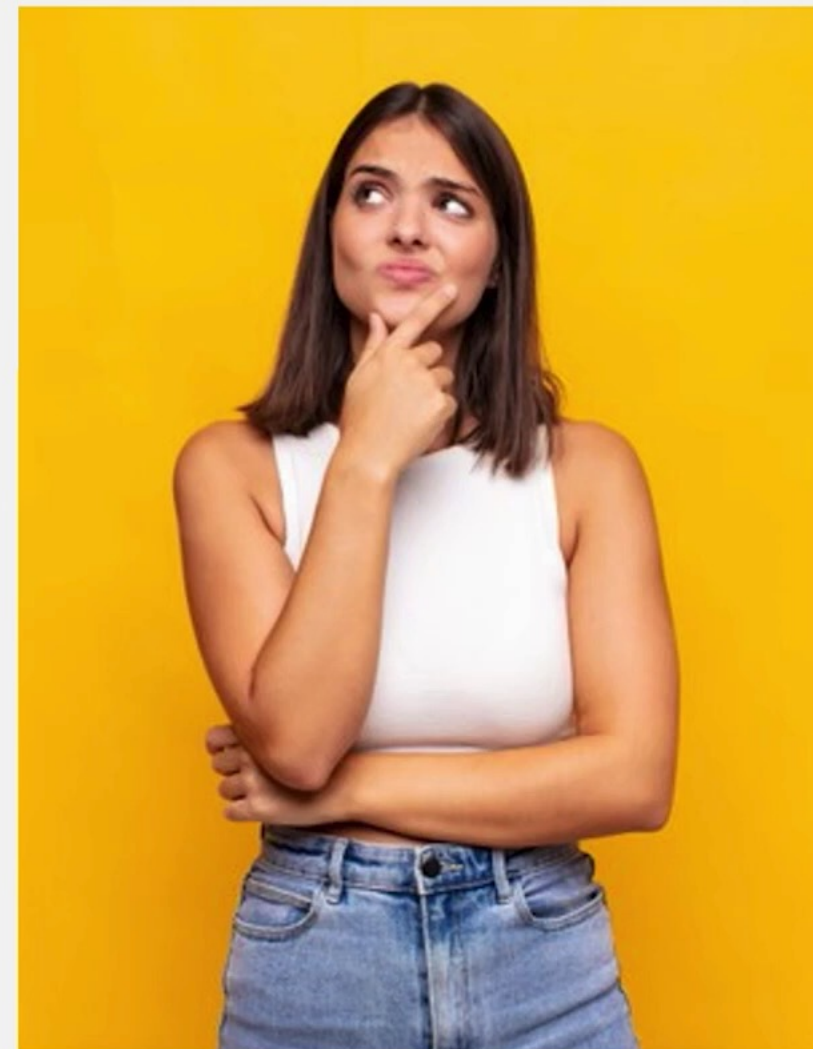
Sprint Restrospective - Retrospectiva

- Sprint Retrospective ocorre ao final de uma Sprint;
- Serve para identificar o que funcionou bem, o que pode ser melhorado e que ações serão tomadas para melhorar;
- Nesta reunião a equipe discute sobre os pontos bons e não tão bons da Sprint, pensando sempre em um plano de ação para melhoria;



Perguntas frequentes

1. Sempre tenho pontos de melhoria? Realmente preciso passar por isso a cada final de Sprint?
2. Posso falar de atitudes de outras pessoas do time? Isso não vai gerar um mal estar?
3. Devo descrever pontos relacionados somente ao DevTeam ou também posso descrever problemas dos demais membros do Scrum Team?(SM, PO, etc.);
4. Como lembrar de todos os pontos que impactaram durante a Sprint?















Como fazer!

Quadro físico

- Ideia de colunas: Funcionou bem, Dentro do esperado, Precisa melhorar;
- Ou com representação simbólica:

Quadro online

- <https://ideaboardz.com/>
- <https://www.funretrospectives.com/>
- <https://miro.com/templates/retrospective-tool/>



Refinamento e MVP

Aula 5

Refinamento do Backlog

- O Objetivo do **Refinamento do Backlog** é aprimorar o Product Backlog (itens candidatos para entrar na Sprint);
- Realizada próximo ao final da Sprint corrente, garantindo assim, que os itens estejam prontos para a próxima Sprint.





Papel do DevTeam

- Deve entender cada um dos itens apresentados pelo PO;
- Deve ajudar o PO a identificar qualquer tipo de problema no detalhamento das histórias;
- Deve indicar alguns insumos que podem inviabilizar a execução da História;
- Ex: integrações com outros times, falta de infraestrutura, documentação de parceiros, protótipos, etc.

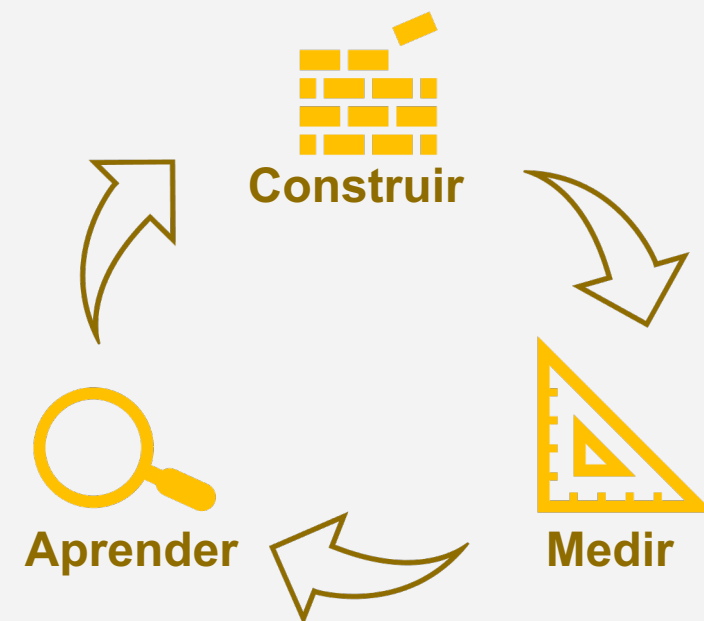


Exemplo de questionamentos:

1. O que deve acontecer se o usuário inserir dados errados aqui?
2. Todo usuário vai ter permissão para acessar essa parte do sistema?
3. O que acontece se o usuário abandonar o sistema?
4. O que acontece se um serviço estiver fora?
5. A arquitetura já configurou o servidor para isso?
6. O time X já fez a integração que necessitamos?

















MVP

- MVP é o acrônimo para **Minimum Viable Product** ou na tradução: Produto Mínimo Viável;
- É um conjunto de testes primários feitos para validar a viabilidade do negócio;
- Produto com o mínimo de recursos possíveis;
- O MVP é aquela versão do produto que permite uma volta completa do ciclo **Construir-Medir-Aprender**, com o mínimo de esforço e o menor tempo de desenvolvimento”.

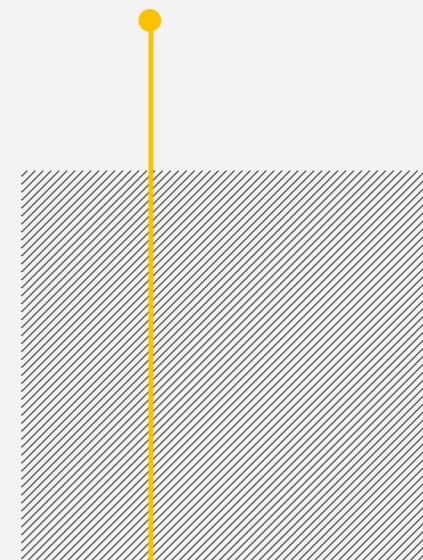


Validar hipótese

O MVP prova a visão inicial, revelando se aquela boa ideia corresponde mesmo um produto interessante (com usuários reais) ou se era apenas uma “expectativa utópica”.

1	2	3	4
			
			
			
			

Exemplo do Spotify



Alguns cases de **sucesso**

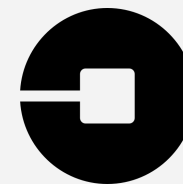
Essas empresas de ramos diferentes usaram o conceito de MVP com ideias simples, com pouco investimento e pouco tempo para se tornar o que são hoje.



facebook



airbnb



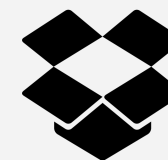
uber



spotify



nubank



dropbox



twitter



groupon



zynga





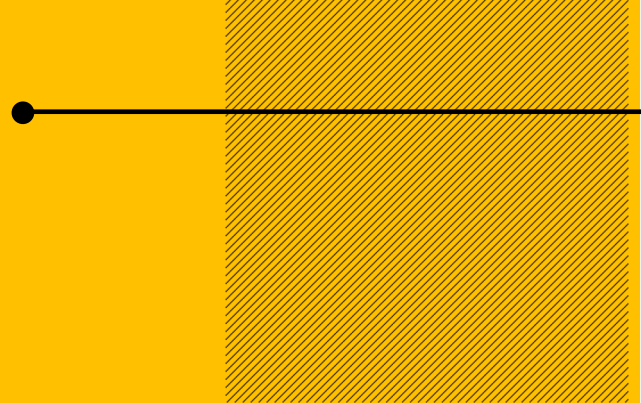
Principais métricas ágeis

Aula 6



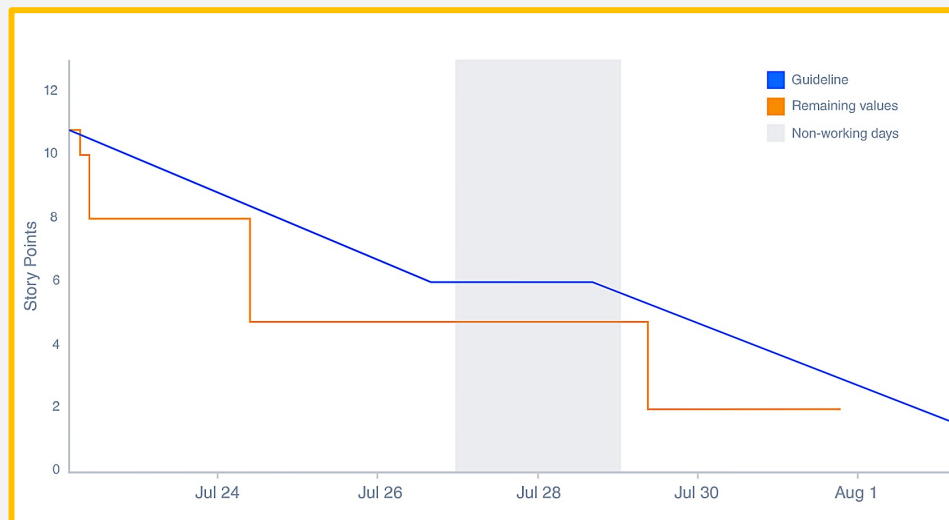
Métricas Ágeis

- Métricas ágeis proporcionam insights sobre a produtividade durante os diferentes estágios de desenvolvimento;
- Ajuda a avaliar a qualidade de um produto e monitorar o desempenho da equipe;
- As principais são:
 - Burndown da sprint
 - Burndow de épico
 - Velocidade
 - Controle
 - Fluxo cumulativo
 - Criado vs Resolvido



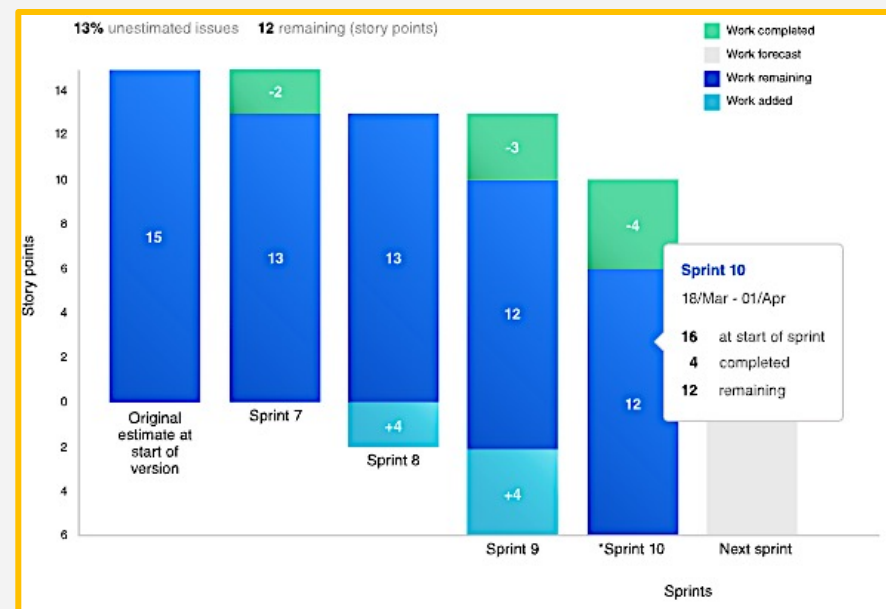
“O que não se
pode medir não se
pode gerenciar”

Gráfico de Burndown



Burndown da Sprint monitora a conclusão do trabalho durante todo a sprint.

Burndow de Épico monitoram o progresso de desenvolvimento de uma grande parte do trabalho do que em relação ao burndown de sprint



Velocidade

Velocidade é a quantidade média de trabalho que uma equipe de scrum conclui durante um sprint, medida em horas ou pontos de história, e é muito útil para previsão.

O PO pode usar velocidade para prever o quão rapidamente uma equipe pode trabalhar em uma lista de pendências.

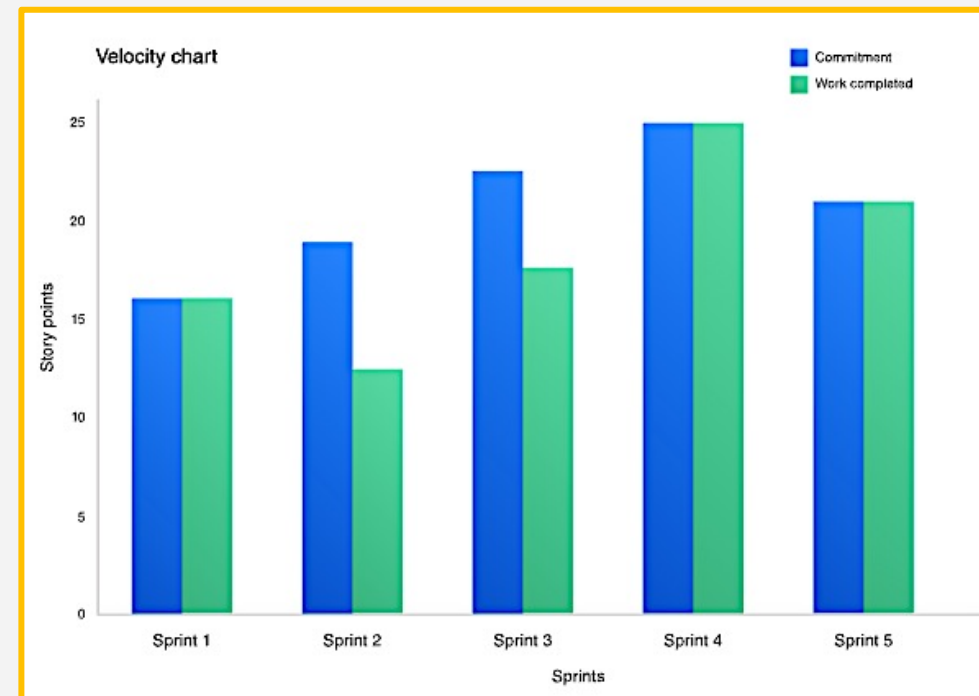


Gráfico de controle

Os gráficos de controle trazem o tempo de ciclo de itens individuais, o tempo total de “em andamento” para “concluído”.

É provável que as equipes com tempos de ciclo mais curtos tenham produtividade maior, e as equipes com tempos de ciclo consistentes por muitos itens são mais previsíveis na entrega de trabalho.

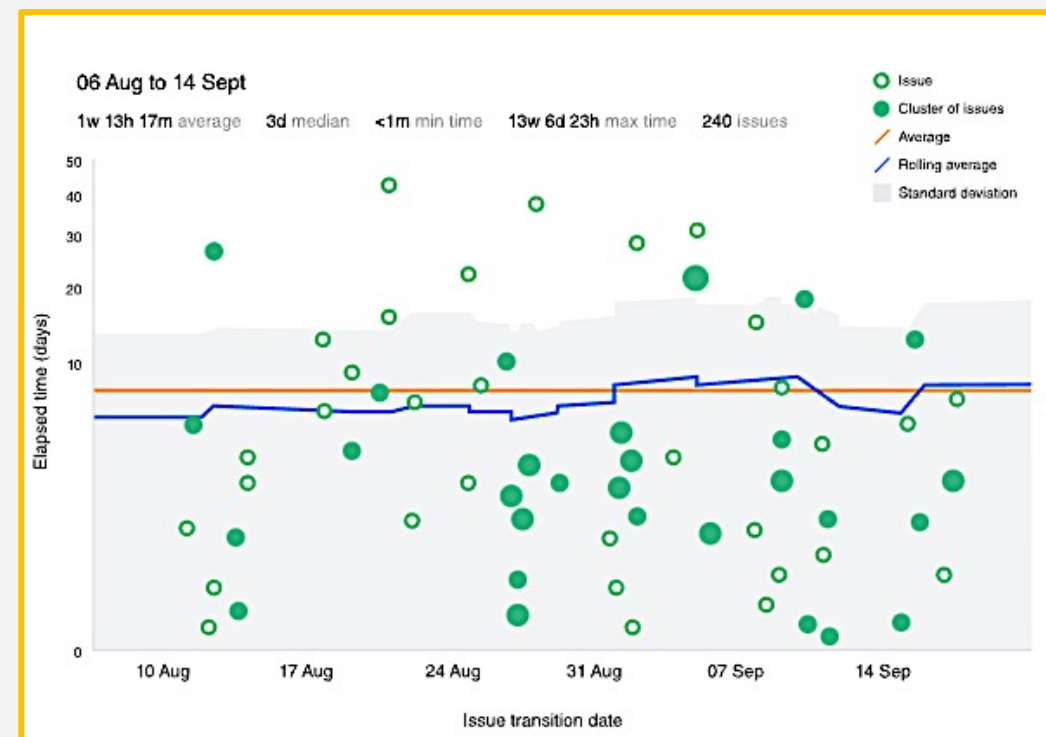
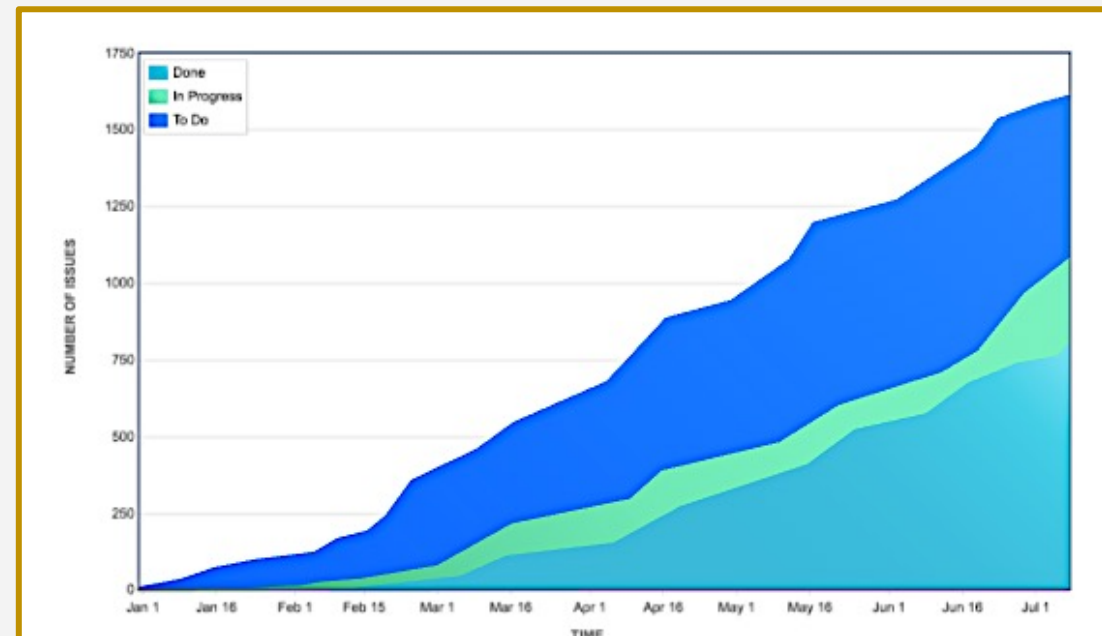


Diagrama de **fluxo cumulativo**

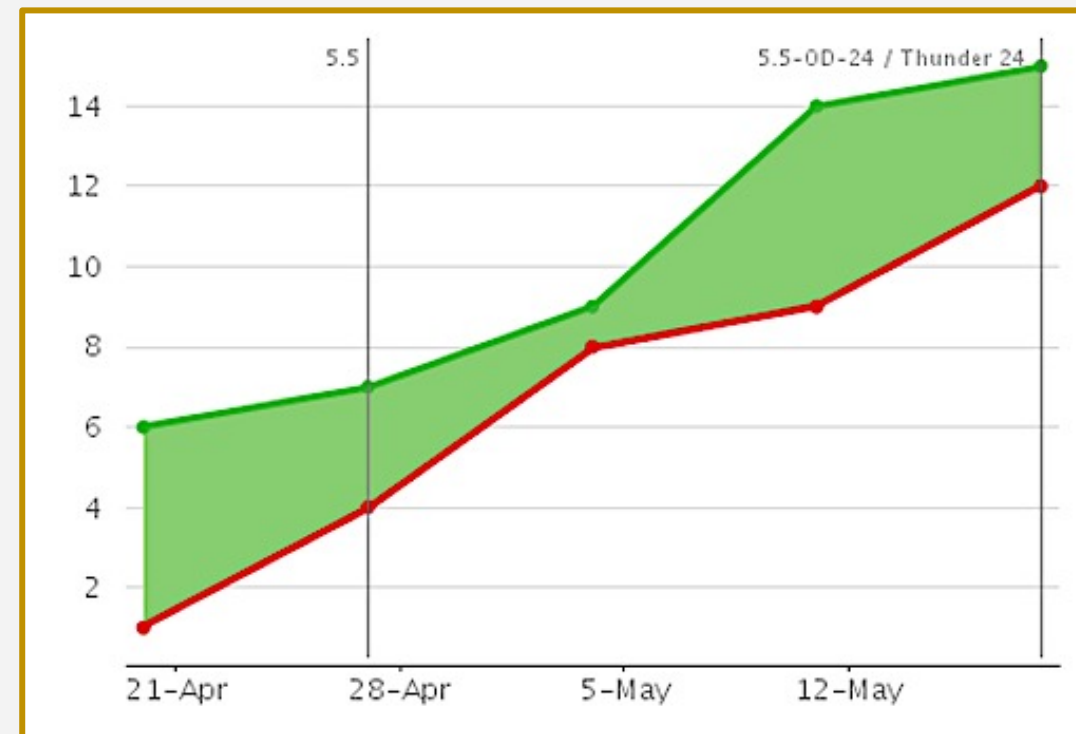
A função deste diagrama é identificar gargalos nos processos, garantindo, assim, a estabilidade do fluxo de trabalho.

O gráfico gerado permite analisar métricas como a quantidade de atividades em andamento naquele momento, a taxa de transferência dessas atividades e, também, quanto tempo duram os ciclos de produção.



Criado vs Resolvido

Criados versus Resolvidos podem ser usados para mostrar a diferença entre o número de pendências criadas e o número de pendências solucionadas ao longo do tempo.



Outras métricas relevantes

- Lead time (tempo de espera)
- Cycle time (tempo de ciclo);
- Desperdício (Ex. Mudança de escopo, Trabalho inacabado, troca de tarefa, etc;
- Cobertura de testes;
- Cobertura de testes de código;
- Reincidência de Bugs;
- Bugs escapados / Incidentes.





Referencias:

- <https://abstartups.com.br/entenda-o-que-e-mvp/>
- <https://www.napratica.org.br/mvp-empresas-tecnologia/>
- <https://www.atlassian.com/br/agile/tutorials/burndown-charts>
- <https://www.heflo.com/pt-br/agil/tudo-sobre-scrum/>
- <https://www.atlassian.com/br/agile/project-management/metrics>
- <https://ezdevs.com.br/times-ageis-conheca-as-principais-metricas/>