

# Metodologias de Trabalho em Equipa

---

## Gestão de Projeto Ágil

# Gestão de Projeto Ágil

---

## Gestão de Projeto com plano vs ágil

- A principal **responsabilidade do gestor de projeto é efetuar a gestão do mesmo** por forma a que o software seja entregue a tempo e dentro do orçamento
- Numa empresa de software, **os gestores precisam de saber:**
  - o que está a acontecer
  - se o projeto tem ou não probabilidade de atingir os objetivos
  - se é possível entregar o software dentro do prazo e do orçamento

# Gestão de Projeto Ágil

---

## Gestão de Projeto com plano vs ágil

- **Abordagem orientada ao plano** (tradicional)
  - Os gestores elaboram um plano para o projeto evidenciando:
    - o que deve ser entregue
    - quando deve ser entregue
    - quem trabalha no desenvolvimento das entregas (deliverables)
  - Exige que o gestor mantenha uma visão estável de tudo o que deve ser desenvolvido e dos processos de desenvolvimento
- **A gestão ágil de projeto requer uma abordagem diferente** que é adaptada ao desenvolvimento incremental e às forças particulares das metodologias ágeis

# Gestão de Projeto Ágil

---

## Gestão de Projeto com plano vs ágil

- **Abordagem ágil**

- As primeiras abordagens ágeis entravam em conflito com a necessidade de visibilidade do negócio
  - equipas auto-organizadas
  - não produzem documentação
  - planeiam o desenvolvimento em ciclos muito curtos
- Como qualquer processo de desenvolvimento de software, o desenvolvimento ágil tem de ser gerido para aproveitar ao máximo o uso do tempo e recursos disponíveis à equipa



- Fornece uma estrutura para a organização de projetos ágeis e visibilidade
- Como o Scrum não é um método para gestão de projetos no sentido convencional, os seus criadores desenvolveram outra terminologia: ScrumMaster

# Gestão de Projeto Ágil

---

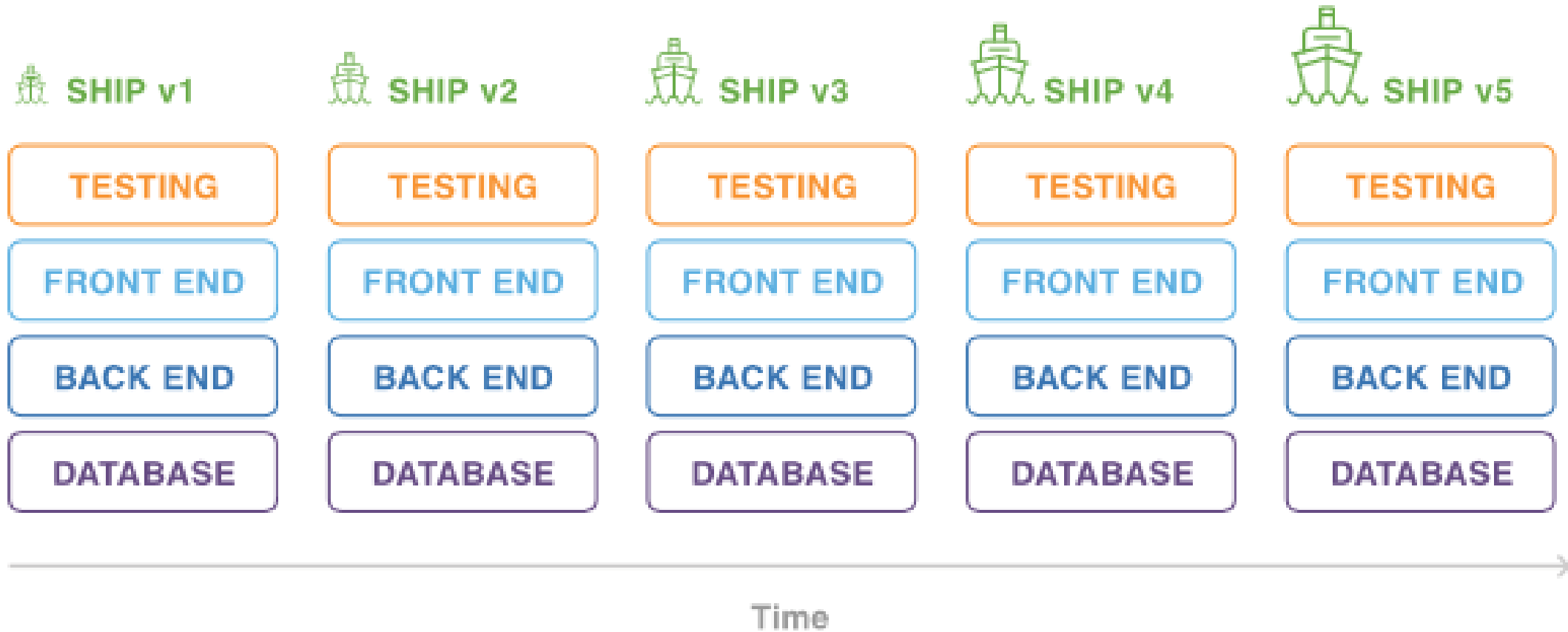
O que é?

- Aplicação de metodologias de gestão ágil de projetos por equipes de software
- **Vantagens**
  - Capacidade de responder às tendências do mercado
  - Velocidade
  - Colaboração
- **Conceitos de gestão ágil**
  - O foco do Product Owner é otimizar o valor do output da equipe de desenvolvimento através da priorização do trabalho
  - A equipe de desenvolvimento só pode aceitar trabalho quando tem capacidade para tal. O Product Owner não faz push para a equipe, a equipe é que faz pull do backlog.

# Gestão de Projeto Ágil

---

O que é?



# Gestão de Projeto Ágil

---

## Programa Ágil

1. **Roadmap** – descreve como um produto se desenvolve ao longo do tempo
  - Composto por **initiatives** – grandes áreas de funcionalidade, e incluem timelines que comunicam quando um recurso estará disponível
  - Pode mudar à medida que o programa se desenvolve
  - O objetivo é manter o roadmap focado nas condições atuais do mercado e nos objetivos a longo prazo
2. **Requisitos** – cada **initiative** divide-se num conjunto de requisitos
  - Os requisitos ágeis são descrições leves da funcionalidade
  - Evoluem com o tempo
  - Refletem o entendimento compartilhado entre a equipa e o produto desejado pelo cliente
  - Os requisitos ágeis permanecem simples e só quando a implementação está prestes a começar é que são detalhados

# Gestão de Projeto Ágil

---

## Programa Ágil

3. **Backlog** – define as prioridades para o programa ágil

- A equipa inclui todos os itens de trabalho no backlog:
  - Funcionalidades
  - Bugs
  - Melhorias
  - Tarefas técnicas ou arquiteturais

4. **Veículos para a entrega** – uso de frameworks (ex., Scum ou Kanban) de entrega de software

5. **Métricas ágeis** – métricas e artefactos que mantêm a equipa focada nos grandes objetivos

- Limites Work In Progress (WIP) – mantêm a equipa focada no desenvolvimento do trabalho com maior prioridade
- Gráficos (ex: Burn-down) – ajudam a equipa a prever a sua velocidade de entrega e a identificar constrangimentos



# Gestão de Projeto Ágil

---

## *Workflow Ágil*

- Qualquer equipa de software usa um processo para completar o seu trabalho
- A normalização desse processo estabelece um **workflow** e permite a sua estruturação e repetição, tornando-o escalável
- É possível definir diversos estados básicos no workflow:
  - To Do
  - In Progress
  - Code Review
  - Done
- Outros:
  - Awaiting QA - à espera da review do tester
  - Ready to Merge - código revisto e pronto para ser merged para o branch master/release

# Gestão de Projeto Ágil

---

## *Workflow Ágil*



### **TO DO**

Trabalho que ainda não foi começado



### **IN PROGRESS**

Trabalho que está em andamento pela equipa



### **CODE REVIEW**

Trabalho completado, mas à espera de ser revisto

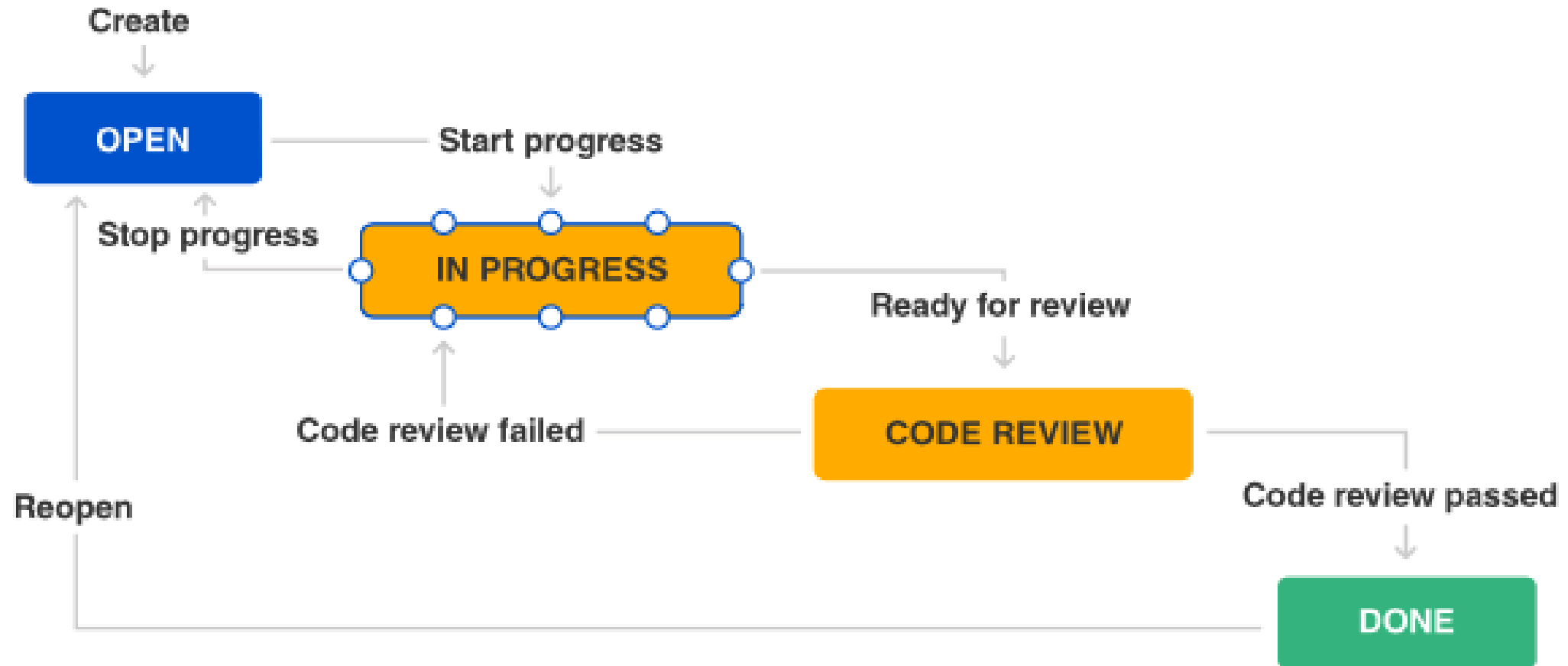


### **DONE**

Trabalho que está totalmente completo e que vai de encontro à definição de DONE da equipa

# Gestão de Projeto Ágil

## Workflow Ágil



# Gestão de Projeto Ágil

---

## *Stories, epics, initiatives e themes*

- O desenvolvimento de um produto requer a estruturação do trabalho: dos objetivos gerais para os detalhes mais pequenos
- A ideia é responder à mudança, reportar o progresso e manter um plano
- Uso das ferramentas:
  - **Stories** – ou User Stories (US), são requisitos pequenos ou pedidos escritos da perspetiva do end user
  - **Epics** – são pedaços grandes de trabalho que podem ser divididos em tarefas mais pequenas (i.e., stories)
  - **Initiatives** – são coleções de epics que vão de encontro a um objetivo comum
  - **Themes** – são grandes áreas de foco da organização

# Gestão de Projeto Ágil

---

*Stories, epics, initiatives e themes*



# Gestão de Projeto Ágil

---

## *Epics vs Stories*

- **Story** – simples narrativa como nos filmes/literatura
  - Stories são algo que a equipa pode fazer commit para terminar num Sprint de 1/2 semanas
- **Epic** – composto por uma série de stories relacionadas e interdependentes
  - Epics existem em menor número e levam mais tempo a terminar
  - Podem ter a duração de 1 mês a 1 trimestre



### EXEMPLO

Objetivo da equipa: Lançar um foguetão para o espaço

# Gestão de Projeto Ágil

---



## *Epics vs Stories*

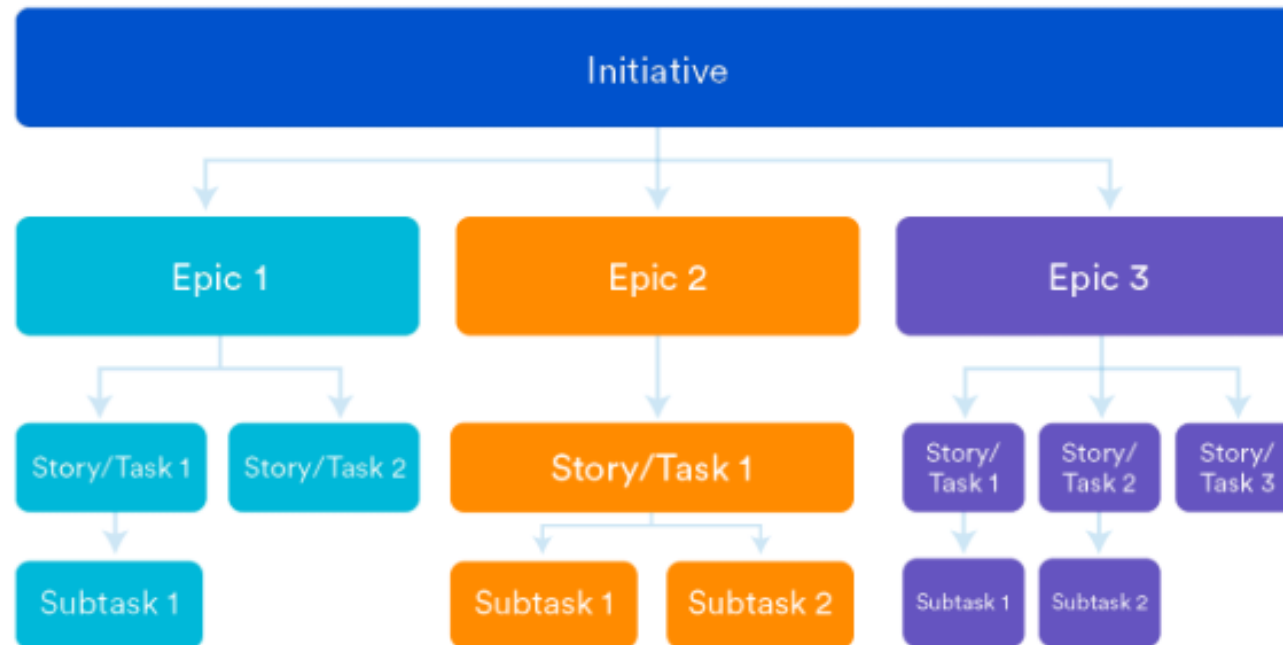
- **Stories:**
  - Utilizadores de iPhone devem aceder à vista vertical da live feed quando usam a app móvel
  - Utilizadores de Desktop devem ter um botão de vista “Fullscreen” no canto inferior direito do video player
  - Utilizadores de Android devem estar ligados à Apple Store
- **Epic:**
  - Melhorar o serviço de streaming para o lançamento Q1

# Gestão de Projeto Ágil

---

## *Epics vs Initiatives*

- **Initiative:**
  - Oferece outro nível de organização acima dos epics
  - Compila epics de diversas equipas para atingir um objetivo maior e mais abrangente
  - Podem ter a duração de vários trimestres a 1 ano





# Gestão de Projeto Ágil

---

## *Initiatives vs Themes*

- **Initiatives:**
  - Têm um design estrutural porque albergam epics, que ao serem atingidos levam ao término da initiative
- **Themes:**
  - São labels que monitorizam os objetivos de alto-nível organizacional
  - Objetivos anunciados pelos fundadores/gestores da organização usualmente anunciados anualmente
  - Ferramenta organizacional que permite colocar labels nos itens do backlog, Epics e Initiatives, de forma a entender qual o trabalho que contribui para determinados objetivos organizacionais
- Theme
  - Safety first



# Gestão de Projeto Ágil

---

## *User Stories*

- No desenvolvimento ágil de software, **as pessoas são colocadas em primeiro lugar**
- As *User Stories* (US) colocam os ***end users* no centro** da conversa
- As **US usam linguagem não técnica** para fornecer contexto para a equipa de desenvolvimento
- A equipa, após ler uma US, **sabe porque é que está a construir** o que tem em mãos e qual o valor que a US cria
- As US ajudam a fornecer uma ***framework* focada no utilizador para o trabalho diário**, gerando colaboração, criatividade e um melhor produto

# Gestão de Projeto Ágil

---

O que são *User Stories*?

- Consistem na **menor unidade de trabalho** da *Framework Agile*
- Uma **US é um objetivo final**, não uma funcionalidade, expressa da perspectiva de software do utilizador
- São frases com **linguagem simples** que descrevem o resultado desejado sem entrarem em detalhes
- Os requisitos são adicionados mais tarde, uma vez acordados pela equipa

# Gestão de Projeto Ágil

---

## Vantagens das *User Stories*

- **Mantêm o foco no utilizador** – uma lista *To Do* mantém a equipa focada nas tarefas que precisam ser feitas, mas uma coleção de US mantém a equipa focada na solução de problemas para utilizadores reais
- **Permitem a colaboração** – com o objetivo final definido, a equipa pode trabalhar em conjunto para decidir qual a melhor forma de servir o utilizador e atingir esse objetivo
- **Impulsionam soluções criativas** – as US incentivam a equipa a pensar de forma crítica e criativa sobre a melhor forma de resolver um objetivo final
- **Criam impulso** – com cada US que passa, a equipa de desenvolvimento desfruta de pequenos desafios e uma pequena vitória, impulsionando o momento

# Gestão de Projeto Ágil

---

Como trabalhar com *User Stories*?

- Uma US é usualmente **escrita pelo Product Owner** e deve ser **integrada no workflow**
- Na **Sprint Planning** a **equipa decide quais as US que são adicionadas ao Sprint Backlog**
  - A equipa discute os requisitos e a funcionalidade que cada US exige
  - Os requisitos são adicionados à US
  - As US são pontuadas com base na sua complexidade ou tempo para conclusão (e.g., sequência de Fibonacci)
  - Uma US deve ser dimensionada para ser concluída num Sprint

# Gestão de Projeto Ágil

---

## Como escrever *User Stories*?

1. **Definição de “Done”** – a US é concluída quando o utilizador pode fazer a tarefa descrita. Certifique-se de que a sua definição está bem descrita.
2. **Delinear sub-tarefas ou tarefas** – decida quais as etapas específicas que precisam de ser concluídas e quem é responsável por cada uma delas
3. **Personas** – para quem? Se houver vários utilizadores finais, considere criar várias US
4. **Etapas ordenadas** – escreva uma US para cada etapa de um processo maior
5. **Ouvir *feedback*** – converse com os utilizadores (clientes) e perceba o problema/necessidade através das suas palavras

# Gestão de Projeto Ágil

---

Como escrever *User Stories*?

6. **Tempo** – muitas equipas de desenvolvimento evitam discussões sobre o tempo, e confiam em estruturas de estimativa

- Como as US devem ser concluídas em apenas 1 Sprint, as US que podem levar semanas ou meses para serem concluídas devem ser divididas em US menores ou serem consideradas no seu próprio Epic



# Gestão de Projeto Ágil

---

## *User Stories* Templates

Title:	Priority:	Estimate:
<b>User Story:</b>  As a [description of user], I want [functionality] so that [benefit].		
<b>Acceptance Criteria:</b>  Given [how things begin] When [action taken] Then [outcome of taking action]		



# Gestão de Projeto Ágil

---

## User Stories Templates

“Como *[persona]*,  
eu *[quero/gostaria que]*,  
*[para quê]*.”

1. “Como ***[persona]***,” – Para quem estamos a construir isto? Não estamos atrás de um título de trabalho, estamos atrás da *persona* da pessoa. Entendemos como a pessoa funciona, como pensa e como se sente.
2. “eu ***[quero/gostaria que]***,” – Aqui descrevemos o intuito, mas não as funcionalidades que eles usam. O que é que eles estão a tentar atingir? Deve ser livre da implementação – se estivermos a descrever algo específico da UI, e não o objetivo do utilizador, estamos a perder o objetivo.
3. “***[para quê]***.” – como é que o desejo imediato deles para fazer algo se integra na *big picture*? Qual o objetivo geral que estão a tentar atingir? Qual o principal problema que precisa de ser resolvido?

# Gestão de Projeto Ágil

---

## Exemplos de *User Stories*

- “Como Ana, eu quero convidar os meus amigos, para que possamos utilizar o serviço juntos.”
- “Como Pedro, eu quero organizar o meu trabalho, para que eu me sinta mais em controlo da situação.”
- “Como gestor, eu quero perceber o progresso dos meus colegas, para que possa melhor perceber o sucesso ou as falhas.”

**Persona + Necessidade + Propósito**

# Gestão de Projeto Ágil

---

## Estimativas

- Fazer **estimativas** é das tarefas mais complicadas para os *developers*
- No desenvolvimento ágil, **é o *Product Owner* que prioriza os itens do *Backlog*** – lista ordenada do trabalho que contém pequenas descrições de todas as funcionalidades para o produto
- Como o *Product Owner* recolhe os requisitos do negócio, **nem sempre entende os detalhes da implementação**
- Uma boa estimativa pode dar ao *Product Owner* uma nova visão do nível de esforço de cada item de trabalho, que em seguida, retroalimenta a sua avaliação da prioridade relativa de cada item

# Gestão de Projeto Ágil

---

## Estimativas

- Quando a equipa de desenvolvimento começa o **processo de estimativa, levantam-se questões sobre os requisitos e US**
- Para o *Product Owner*, **partir os itens de trabalho em pedaços mais pequenos e ter estimativas através de *story points***, ajuda-o a priorizar todas as áreas de trabalho
- No final, o *Product Owner* volta a ordenar os itens do *Backlog*

# Gestão de Projeto Ágil

---

## *Story Points vs Tempo*

- Equipas de software
  - Tradicionais – estimam em formato de tempo (dias, semanas, meses)
  - Ágeis – story points (SP)
- Os Story Points pontuam o esforço do trabalho no formato da sequência de Fibonacci:
- São estimativas do esforço necessário para terminar cada item do Backlog
- A abstração dos SP permite puxar pela equipa para tomar decisões à volta da dificuldade do trabalho



0	1/2	1	2	3	5
8	13	20	40	100	?

# Gestão de Projeto Ágil

---

## Vantagens dos *Story Points*

- As datas não têm em conta o trabalho não relacionado com o projeto que inevitavelmente aparece (e-mails, reuniões e entrevistas)
- Cada equipa estimará o trabalho numa escala ligeiramente diferente, o que significa que a sua velocidade (medida em pontos) será naturalmente diferente
- Após concordância em relação ao esforço relativo de cada valor de SP, a atribuição de SP é rápida e sem grande debate
- Os SP recompensam os membros da equipa por resolverem problemas com base na dificuldade e não no tempo gasto – mantém a equipa focada no valor da entrega, não gastando tempo

# Gestão de Projeto Ágil

---

## *Story Points Planing Poker*

### ***Planning Poker***

1. A equipa analisa um item do *Backlog* discutindo brevemente e cada membro pensa mentalmente uma estimativa
  2. Cada um segura num cartão com o número que reflete sua estimativa
  3. Se todos estiverem de acordo, ok!
  4. Caso contrário, dedicam alguns minutos para entender a lógica por trás de diferentes estimativas
  5. Mas a estimativa deve ser uma atividade de alto nível
- 
- Exemplo: Se uma US-1 tiver 20 SP e a US-2 tiver 40, significa que esta última requer o dobro do esforço que a primeira

# Gestão de Projeto Ágil

---

## Métricas

- Estar no caminho certo durante o projeto significa obter e analisar alguns dados ao longo do tempo
- É importante monitorizar:
  - **Métricas de negócio** – focam-se no atendimentos da solução às necessidades do mercado
  - **Métricas ágeis** – medem aspetos do processo de desenvolvimento
- Para cada Initiative do roadmap, incluir Key Performance Indicators (KPI) de forma a mapearem os objetivos



# Gestão de Projeto Ágil

---

## Métricas

- Incluir critérios de sucesso para cada requisito de produto, tais como a taxa de adoção pelos utilizadores finais ou percentagem de código coberto por testes automáticos.
- Estes critérios de sucesso alimentam as métricas ágeis do programa -> quanto mais as equipas aprenderem, melhor elas se adaptam e evoluem

# Gestão de Projeto Ágil

---

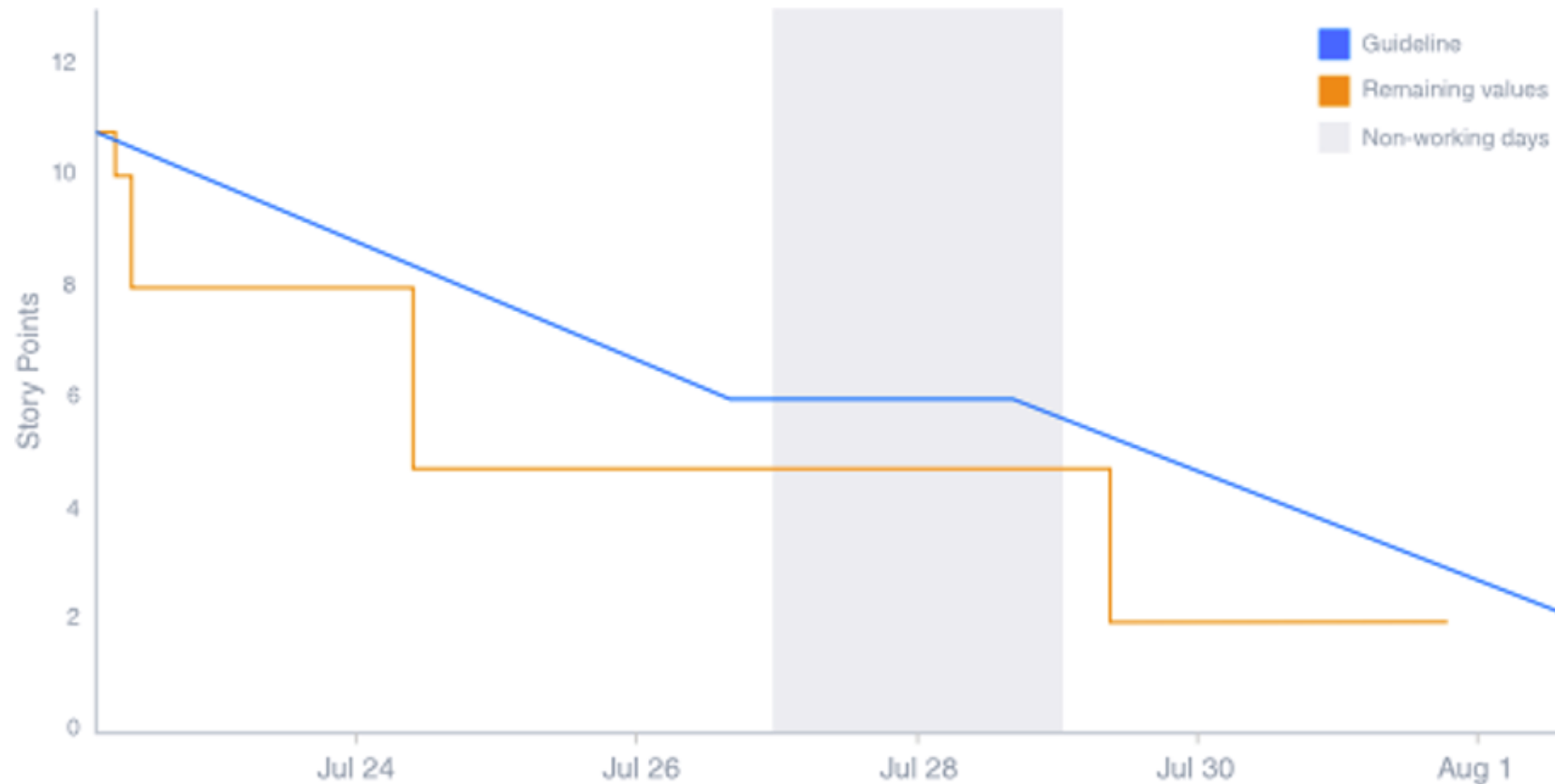
## Métricas: *Sprint Tracking Mechanisms*

- Ferramentas que permitem que o projeto esteja no bom caminho para fornecer a solução esperada dentro do planeamento desejado
- **Métricas ágeis:**
  - Burn-Down chart
  - Burn-Up chart
  - Velocity chart
  - Cumulative flow diagram

# Gestão de Projeto Ágil

## *Burn-down chart*

- **Burn-down** – monitoriza o trabalho concluído (normalmente por dia) em relação à taxa de conclusão projetada para a *release (sprint)* atual do projeto



# Gestão de Projeto Ágil

---

## *Burn-down chart*

- Ferramenta de representação visual de medição que mostra o trabalho que resta no Sprint (usualmente em *story points*)
- Geralmente é atualizado na *Daily Scrum*
- À medida que o Sprint progride, a quantidade de trabalho restante deve diminuir constantemente e tender a ser concluída no último dia do Sprint
- Burn-downs que mostram o trabalho a crescer ou com poucas tarefas concluídas são sinais para o Scrum Master e toda a equipa de que o Sprint não está bem encaminhado

# Gestão de Projeto Ágil

---

## *Burn-up chart*

- **Burn-up** – mostra o *scope* do projeto e do trabalho concluído ao longo do tempo
- **Eixo vertical:** Quantidade de trabalho (*story points*)
- **Eixo horizontal:** Tempo (dias)
- **Team effort** – progresso da equipa (SP concluídos) em relação ao *sprint backlog*
- **Total effort** – esforço total necessário para atingir o objetivo da equipa (*sprint/release*)
- **Ideal burnup** – ritmo de trabalho ideal necessário para cumprir o prazo do projeto

# Gestão de Projeto Ágil

## *Burn-up chart*



# Gestão de Projeto Ágil

---

## *Burn-up chart*

- O *Burn-up* é atualizado no final de cada unidade de tempo e mostra o progresso real da equipa
- A linha de *team effort* sobe de baixo para cima em direção à linha de *total effort* – quando as duas linhas se encontram, o objetivo da equipa é atingido
- A equipa pode ver o trabalho restante pela observação da distância entre a linha de *team effort* e a linha de *total effort*
- **Vantagem do *burn-up*:**
  - Simplifica a monitorização, a modificação e melhora o processo de trabalho da equipa

# Gestão de Projeto Ágil

---

## *Burn-Down vs Burn-Up Chart*

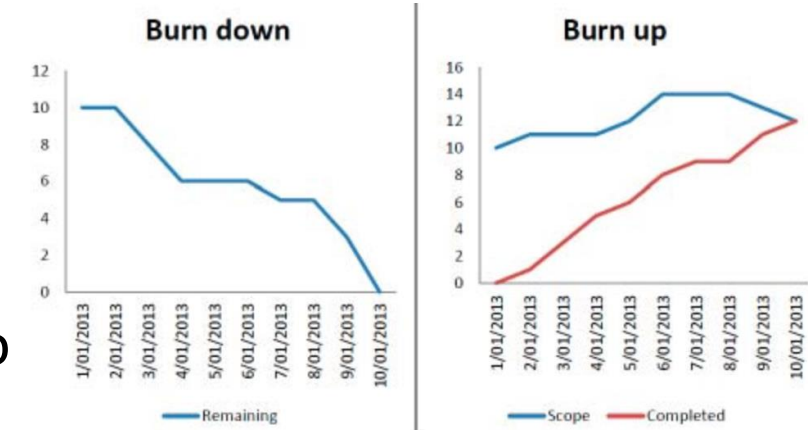
- **Burn-down chart**

- Mostra a quantidade de trabalho restante num projeto
- A linha é decrescente
- Mais simples de entender porque só tem 1 linha

- **Burn-up chart**

- Mostra a quantidade de trabalho que foi concluída e o scope total do projeto
- A linha é crescente
- É mais completo porque mostra o trabalho atingido e o trabalho concluído nos incrementos anteriores

- Ambos usam os mesmos eixos



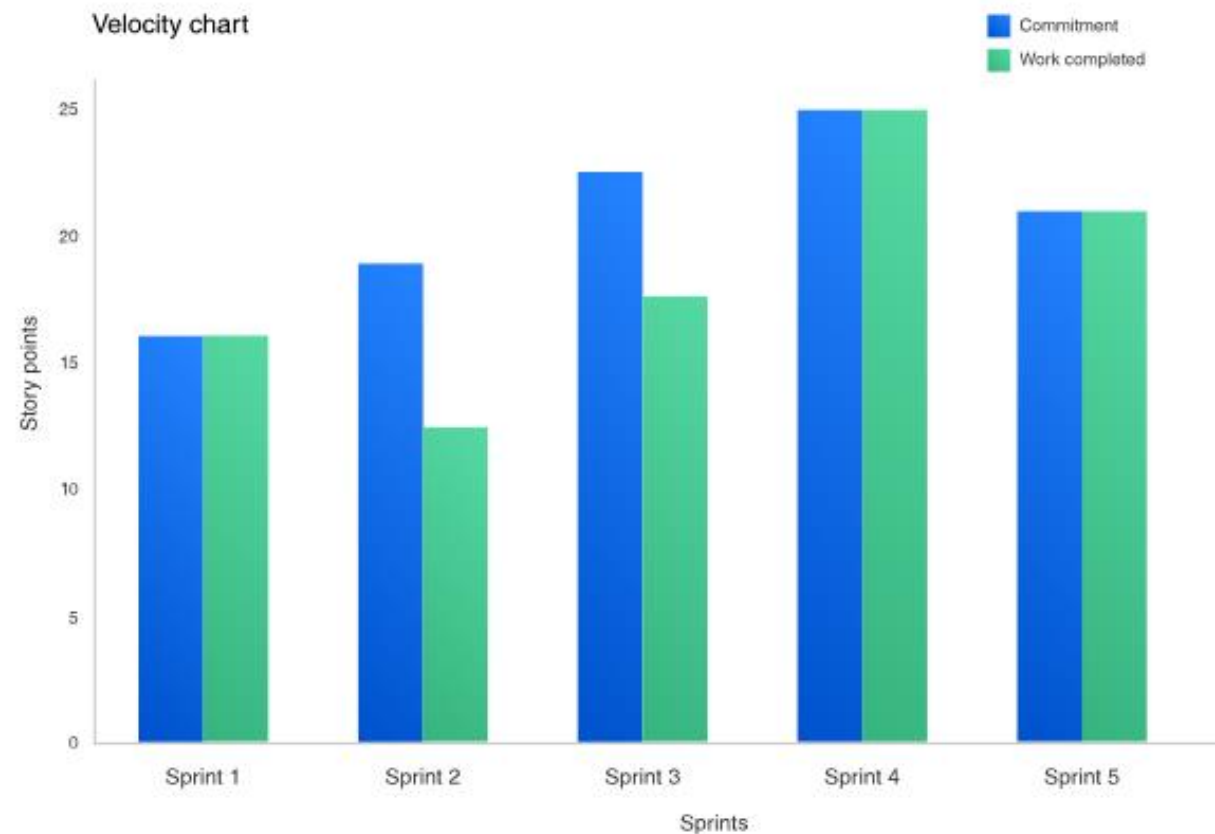


# Gestão de Projeto Ágil

---

## *Velocity Chart*

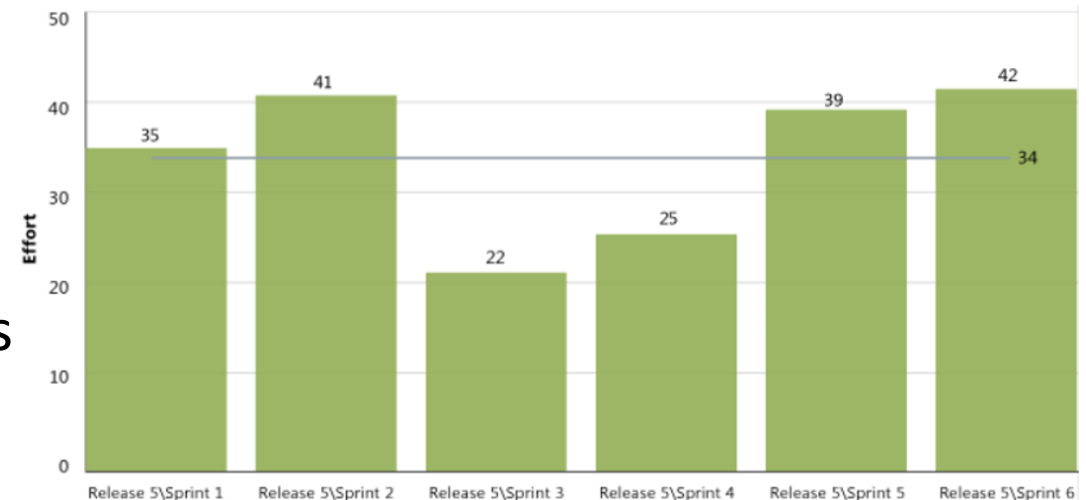
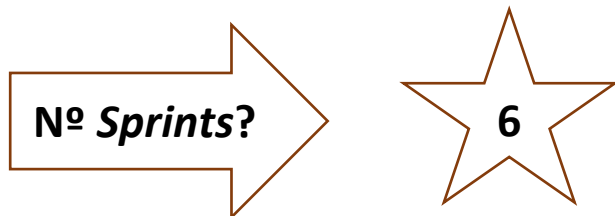
- **Velocity** – quantidade média de trabalho que a equipa conclui durante um *Sprint*, medido em *story points* ou em horas, muito útil para a previsão



# Gestão de Projeto Ágil

## Velocity Chart

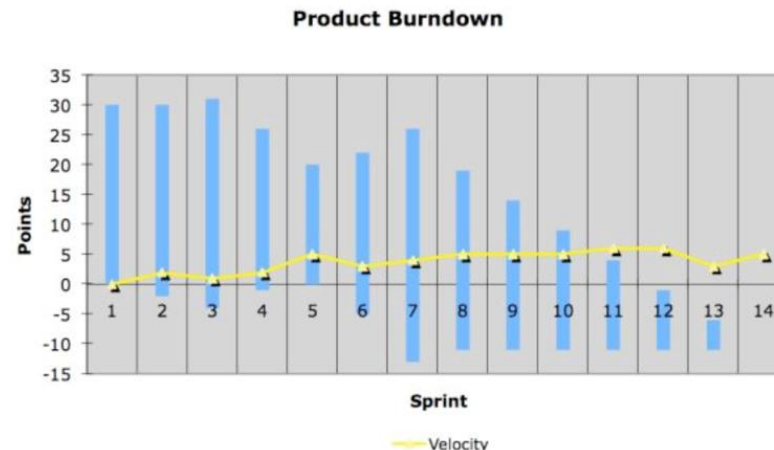
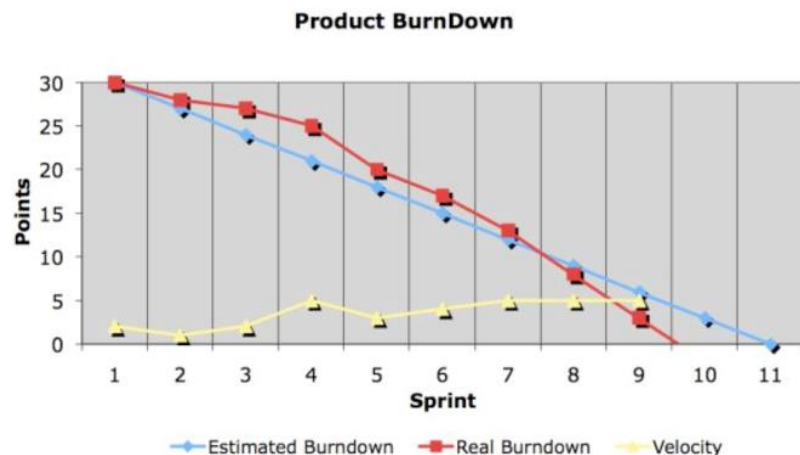
- Regra para calcular a velocidade: apenas as stories concluídas no final da iteração são contadas. É estritamente proibido contar trabalho parcialmente concluído (p. ex., apenas código mas sem ter sido testado)
- Após alguns Sprints, a velocidade de uma equipa é previsível e permite uma estimativa bastante precisa sobre o tempo necessário até que todas as entradas no Product Backlog sejam concluídas
- Exemplo:
- **Velocidade da equipa** = 30 story points
- **Quant. restante de trabalho** = 155 story points



# Gestão de Projeto Ágil

## *Burn-Down vs Velocity*

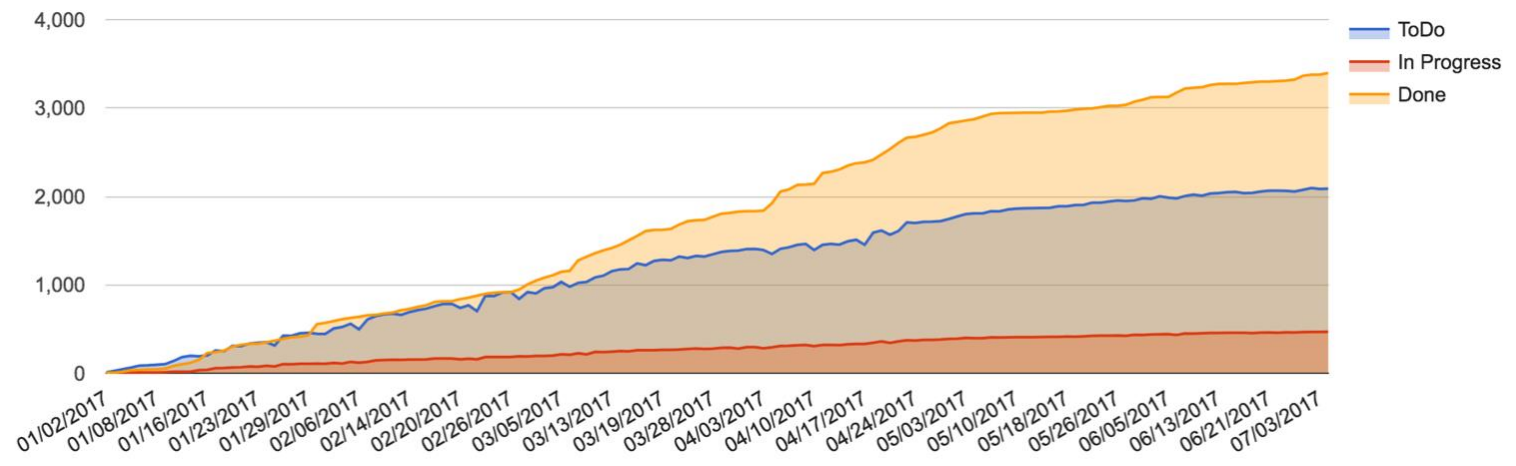
- Na prática, as entradas no *Product Backlog* são alteradas ao longo da duração do projeto. Novas *stories* são adicionadas e outras são alteradas/eliminadas
- No gráfico *Burndown* simples, a velocidade da equipa e a mudança no *scope* não são diferenciadas
- No gráfico de Velocity, o tamanho de cada barra representa a quantidade total de trabalho restante no início de cada sprint
- A velocidade da equipa é subtraída do topo, enquanto as mudanças no *scope* alteram a parte inferior da barra



# Gestão de Projeto Ágil

## Comulative Flow Chart

- **Comulative Flow** – gráfico de área que mostra o progresso dos itens de trabalho de um projeto por um período de tempo específico
- Ferramenta fundamental para visualizar o progresso do projeto e ajuda a identificar possíveis problemas
- Mostra a contagem e ritmo de progresso de itens do *Backlog* no período selecionado
- É uma versão avançada do gráfico *Burn-up*



# Gestão de Projeto Ágil

---

## Métricas

- Outras métricas:
  - Frequência e velocidade de entrega
  - Frequência das releases para produção
  - Quanto tempo é necessário para a equipa resolver uma correção de emergência para a produção?
  - Quantos defeitos são encontrados:
    - durante o desenvolvimento?
    - após a release entregue aos clientes?
    - por pessoas de fora da equipa?
  - Quantos defeitos são adiados para uma release futura?
  - Quantos pedidos de suporte ao cliente estão a chegar?
  - Qual é a percentagem de cobertura de testes automáticos?

# Gestão de Projeto Ágil

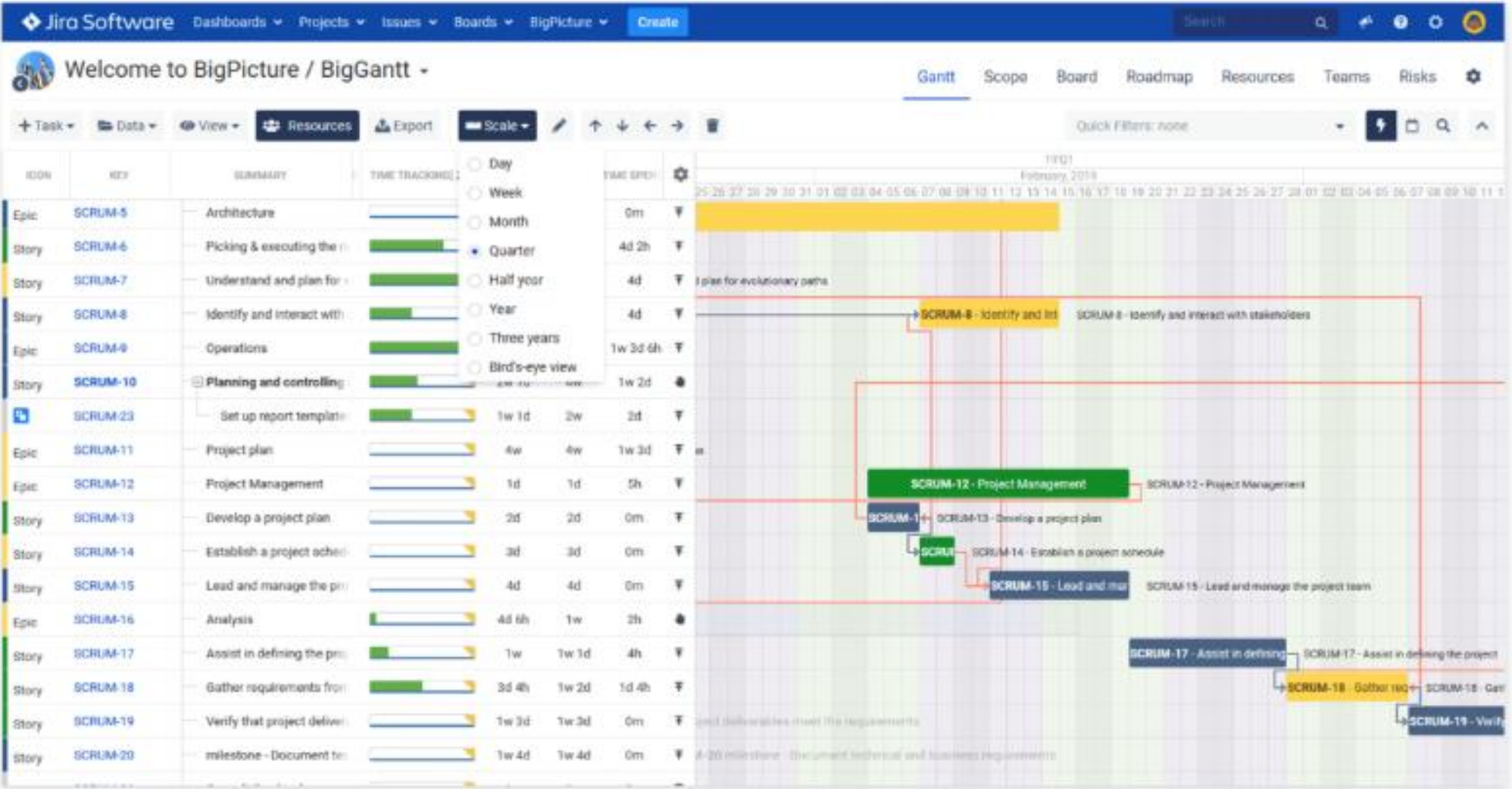
---

## Diagrama de Gantt

- Diagrama de Gantt – gráfico de barras horizontal baseado no tempo que representa o planejamento do projeto
  - Cada tarefa listada tem a sua barra representada no tempo
  - Componentes:
    - Datas de início e de fim
    - Milestones
    - Dependências entre tarefas
- Benefícios
  - Simplifica projetos complexos
  - Representa toneladas de dados de uma forma incrivelmente visual e agregada
  - Vista única sobre a totalidade do projeto com milestones e deadlines

# Gestão de Projeto Ágil

## Diagrama de Gantt



# Gestão de Projeto Ágil

---

## Diagrama de Gantt

- Um diagrama de Gantt pode ser usado por uma gestão do tipo Waterfall ou híbrida para:
  1. **Determinar o planeamento do projeto** – quebrar o projeto em pequenos pedaços de trabalho que é possível gerir; planejar os epics, stories, tarefas e sub-tarefas
  2. **Estabelecer papéis, responsabilidades e recursos** – assegurar que existem recursos suficientes para a quantidade de trabalho existente
  3. **Monitorizar o progresso do projeto** – usar barras de progresso desde as sub-tarefas até aos epics
  4. **Identificar milestones** – “momentos da verdade” que são atingidos pelas equipas
  5. **Encontrar e reportar problemas** – localizar problemas reais e usar o diagrama para identificar as tarefas que vão afetar a data final do projeto



## Referências bibliográficas

---

- Sommerville, I. (2015). “Software Engineering”, 10th Edition, Addison-Wesley.  
<http://dinus.ac.id/repository/docs/ajar/Sommerville-Software-Engineering-10ed.pdf>
- Santos, D. (2021). Conteúdos lecionados na UC de IAPSI do TeSP PSI. IPLeiria.
- Drumond, C. “Agile Project Management - How agile methodologies can work for your software team”. <https://www.atlassian.com/agile/project-management>