

FIAP

FIAP

SLIDER ▶▶▶



# SOFTWARE ENGINEERING AND BUSINESS MODEL

Prof. M.Sc. Felipe Desiglo Ferrare  
[proffelipe.ferrare@fiap.com.br](mailto:proffelipe.ferrare@fiap.com.br)

# Introdução ao Gerenciamento de Projetos III

Prof. M.Sc. Felipe Desiglo Ferrare  
[proffelipe.ferrare@fiap.com.br](mailto:proffelipe.ferrare@fiap.com.br)

## Entrega Contínua e DevOps

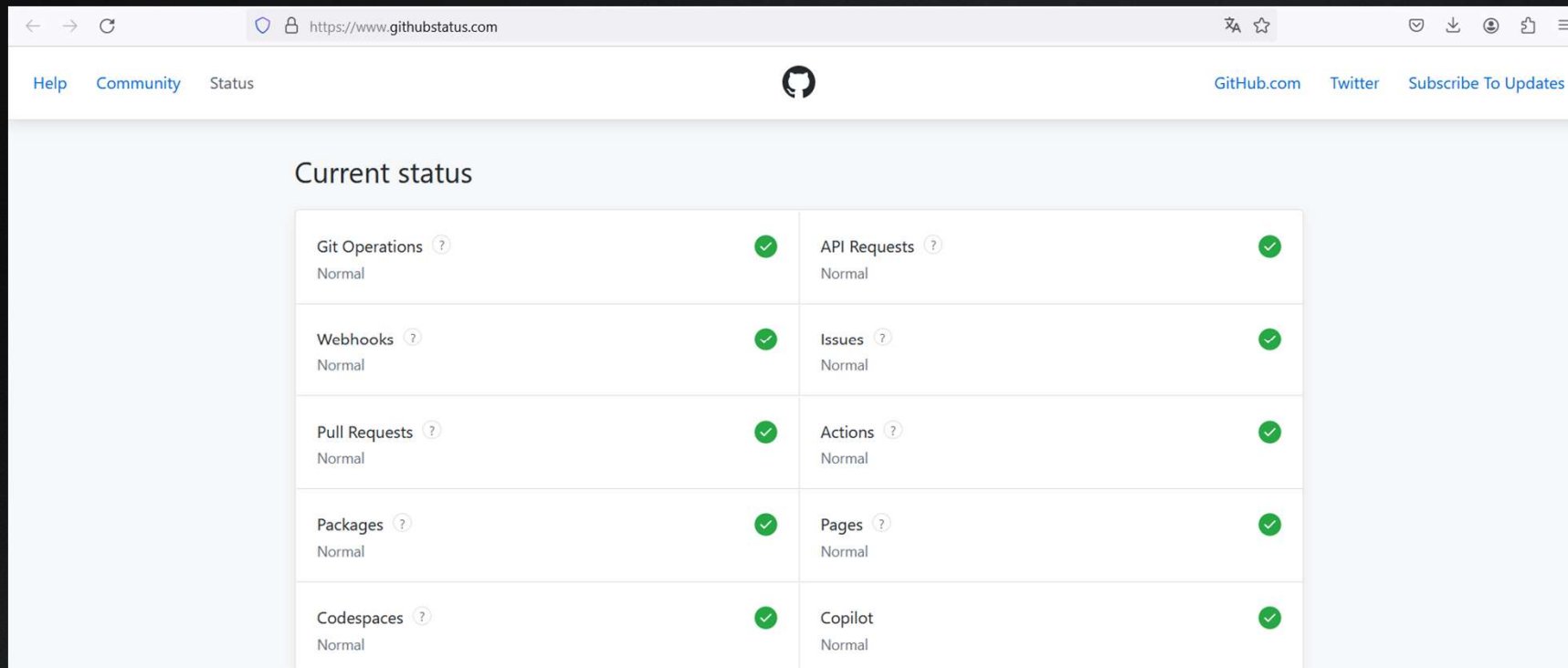
- **DevOps** é a cultura que visa unir os esforços de times de Desenvolvimento e Operação
- **Integração Contínua (CI)** é a integração de novas features ao código e validação da integração
- **Entrega Continua (CD)** é a entrega de código ou funcionalidades na menor janela tempo possível (uma sprint ou menos)





# Entrega Contínua e DevOps

- Assim como no Kanban que temos um quadro, podemos criar interfaces em sites (públicos/privados) ou quadros para ver status



The screenshot shows the GitHub Status page at <https://www.githubstatus.com>. The page displays the 'Current status' of various GitHub services, all of which are currently 'Normal' and marked with a green checkmark.

Service	Status
Git Operations	Normal
Webhooks	Normal
Pull Requests	Normal
Packages	Normal
Codespaces	Normal
API Requests	Normal
Issues	Normal
Actions	Normal
Pages	Normal
Copilot	Normal

## Controle de Trabalho Burndown

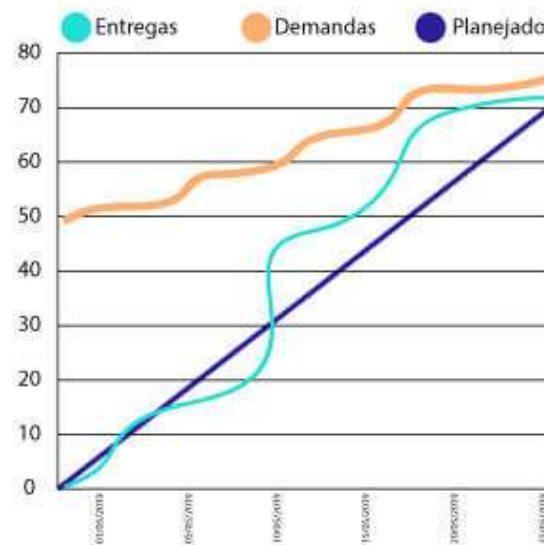
- Monitorar trabalho Feito (Atingir Meta do Sprint)



## Controle de Trabalho Burnup

- Monitorar Projeto (Features sendo entregues)

Gráfico de burnup





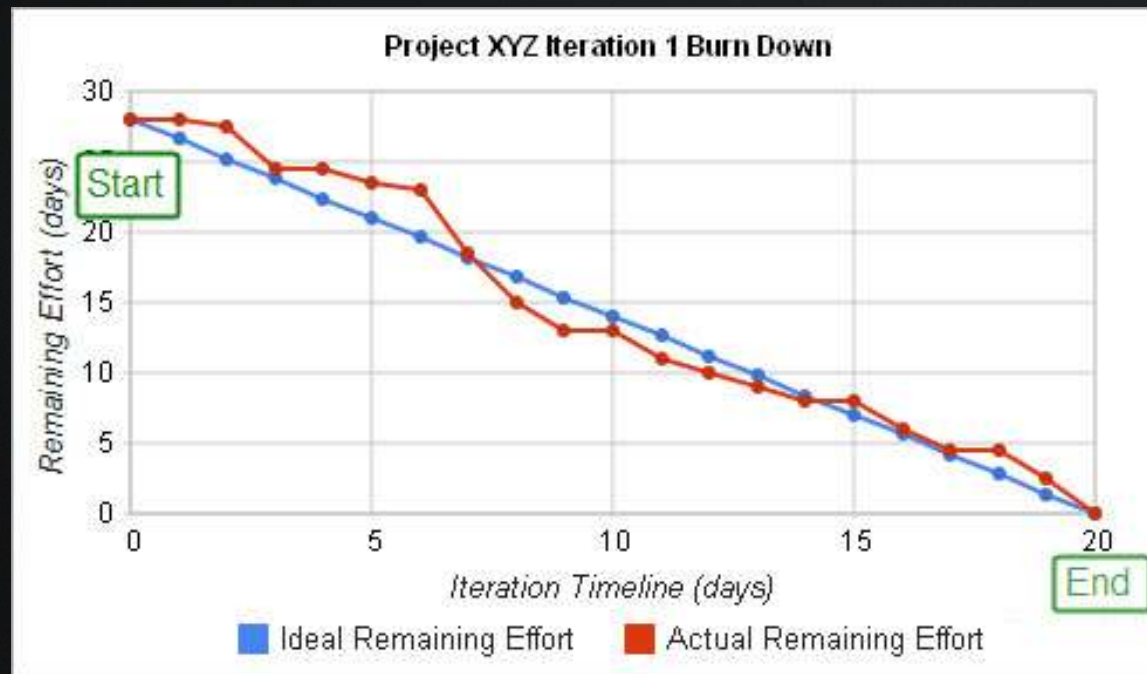
## Controle de Trabalho

### Burndown e Burnup

- Monitorar trabalho e progresso
- 2 linhas uma é a base (planejado) e a outra o real (Feito efetivamente)

## Controle de Trabalho Burndown

- Atualizado diariamente pode ser exibido junto ao quadro Kanban



## Transparência

- Transparência é um dos objetivos do Scrum
- Saber o status do trabalho o que as pessoas fazem e os problemas, não para controle, mas para colaboração !

## Requisitos

- O que temos que fazer
- Coletar e refinar e escrever requisitos
- Requisitos vão construir nossos Épicos que vamos quebrar em Funcionalidades e Tarefas no Backlog



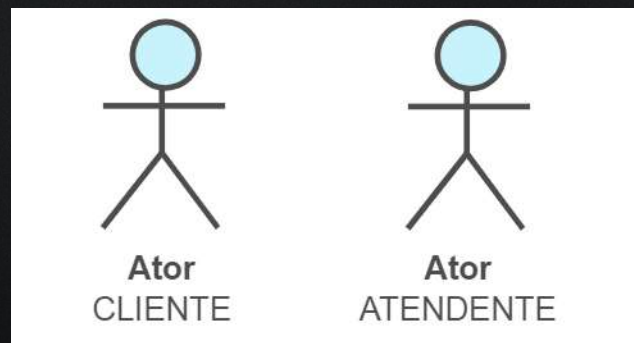
## DIAGRAMA DE CASO E USO

- Por meio dele, documentamos o sistema sob o ponto de vista do usuário, formalizando o que ele pode ou não fazer. É elaborado durante as reuniões entre a equipe de desenvolvimento do sistema e as partes interessadas (stakeholders), com o objetivo de especificar os requisitos. É composto do **Diagrama de Caso** de Uso e da **descrição dos casos de uso** (usualmente, uma descrição textual).

## DIAGRAMA DE CASO E USO

### Ator

- Um ator corresponde a um papel representado por algo ou alguém, sendo qualquer elemento externo ao sistema. Pode ser um ser humano, hardware, dispositivo ou o sistema externo que interage com o sistema em questão.
- O ator é quem interagirá com o sistema, e por “interagir” entendemos que o ator trocará informações com o sistema (enviará informações para processamento ou receberá informações processadas provenientes do sistema).



## DIAGRAMA DE CASO E USO

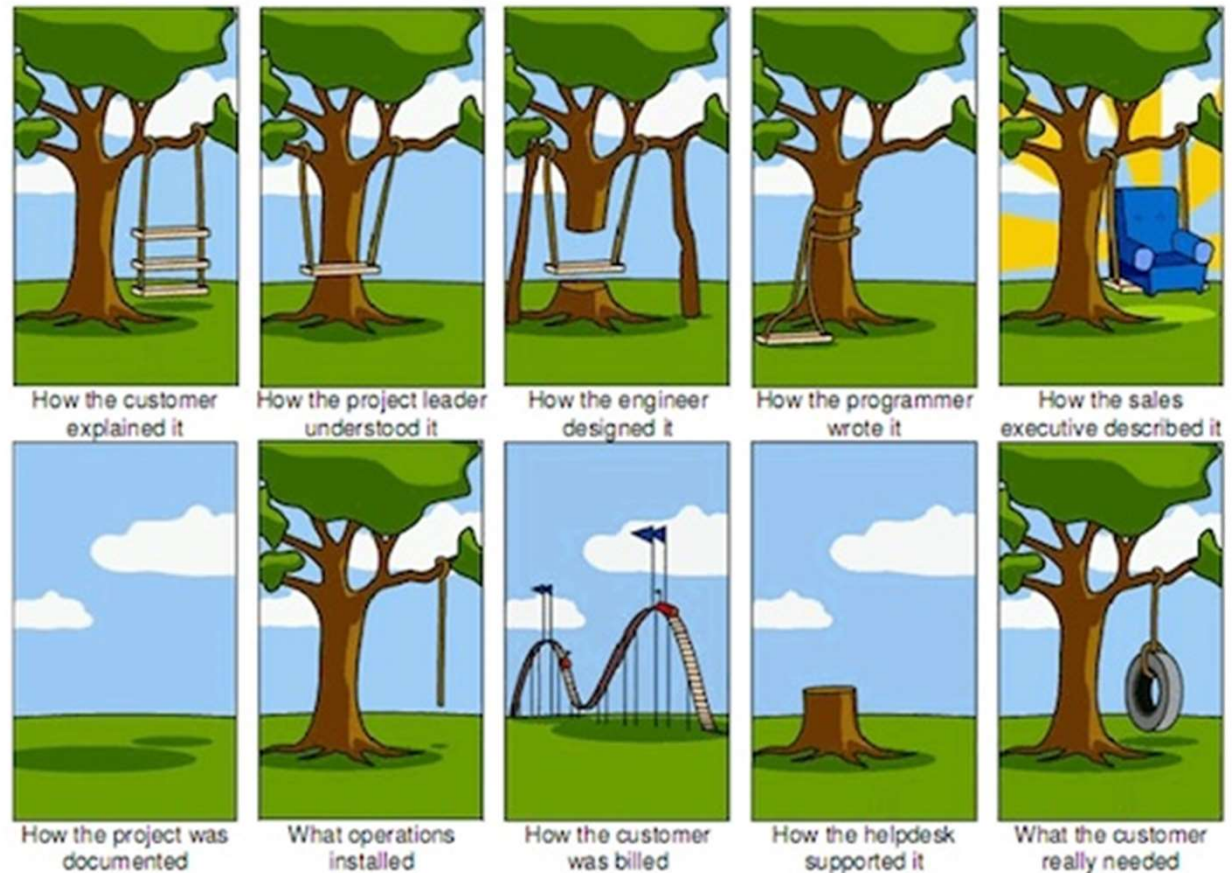
Item	Valor
Caso de uso	UC01 - Comprar cerveja.
Sumário	Permite que o usuário efetue a compra de cerveja.
Ator	Cliente
Precondição	Ter estoque disponível.
Pós-condição	Registrar a compra e o pagamento.
Fluxo principal	FP01 - O cliente seleciona a cerveja que deseja comprar. FP02 - O cliente informa a quantidade desejada. FP03 - O sistema verifica se há estoque disponível. FP04 - O sistema calcula o valor total da compra.
Fluxo(s) alternativo(s)	FA001 - O cliente pode alterar a quantidade desejada. O sistema retorna ao FP03.
Fluxo(s) de exceção	FE01 - O sistema exibe a mensagem: "Cerveja indisponível". Encerra o caso de uso.



## Requisitos

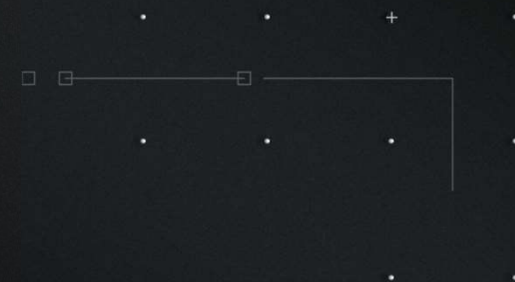
- Mesma imagem dos requisitos
  - Pessoas diferentes
- tem diferentes visões do produto

-> Necessário colocar as expectativas de forma alinhada entre todos os stakeholders





Obrigado!



FIAP