



iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA



emprego
digital

Módulo 3: Princípios de Desenvolvimento de Software

Aula 2

Metodologias de Desenvolvimento



Desenvolvimento de Software

As várias fases de desenvolvimento de software desde a especificação inicial à fase final de manutenção podem ser modeladas para corresponder a representações simplificadas de perspectivas:

- **Workflow** - Diagramas de estados, atividade e sequência do software;
- **Dataflow** - Fluxo da informação pelo sistema;
- **Role/Action** - Quem faz / O quê;

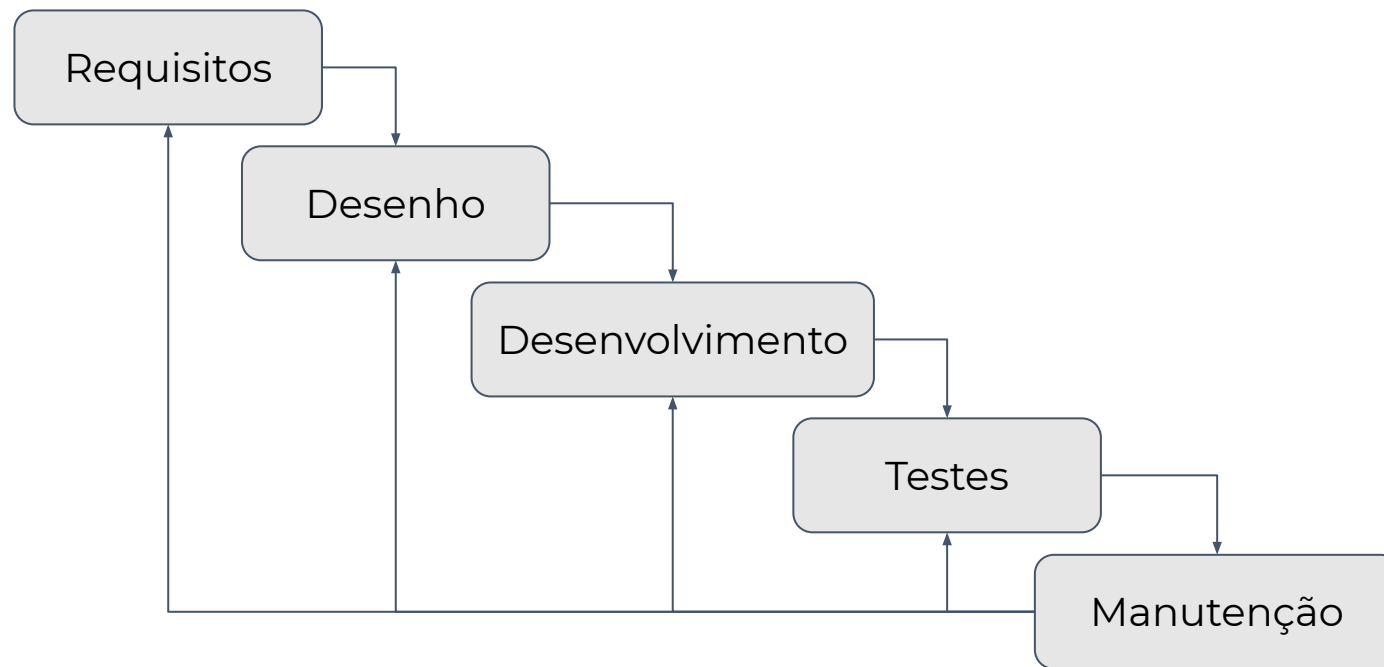
Modelos de Desenvolvimento

- Surgiram da necessidade de aumentar a performance do serviço do desenvolvimento de software;
- Prevêem aumentar a colaboração dentro das equipas;
- São filosofias e formas de pensar.



Modelo *Waterfall*

O modelo de desenvolvimento de software mais antiquado. Baseado nos modelos de desenvolvimento de outras engenharias.



Waterfall - Vantagens

- Estrutura clara e concisa;
- Fácil para projetos pequenos em que se consegue delimitar concretamente um início e um fim;
- Previsível dados requisitos e documentação bem definida;
- Alta visibilidade: Passos bem definidos que permitem ao cliente ver o avanço no desenvolvimento;
- Passagem de informação sistemática e sem perdas.

***Waterfall* - Desvantagens**

- Em projetos de longa duração torna-se impraticável;
- Não permite realizar alterações a meio do desenvolvimento;
- Exclui em grande parte o cliente/end-user da equação;
- Testes só são feitos no fim, o que leva a um ciclo de vida instável.

Modelo *Agile*

Modelo alternativo ao *Waterfall* que providencia uma abordagem realista ao desenvolvimento de software.

Depende dos requisitos iniciais e da ideia do produto final.



Agile - Vantagens

- Ótimo para projetos longos devido à flexibilidade nas alterações e no constante feedback;
- Separação do ciclo de desenvolvimento em “*sprints*” que tornam o processo mais rápido;
- Foco maior na qualidade do produto/serviço final;
- Possível testar em qualquer fase;
- Altamente cooperativo.

Agile - Desvantagens

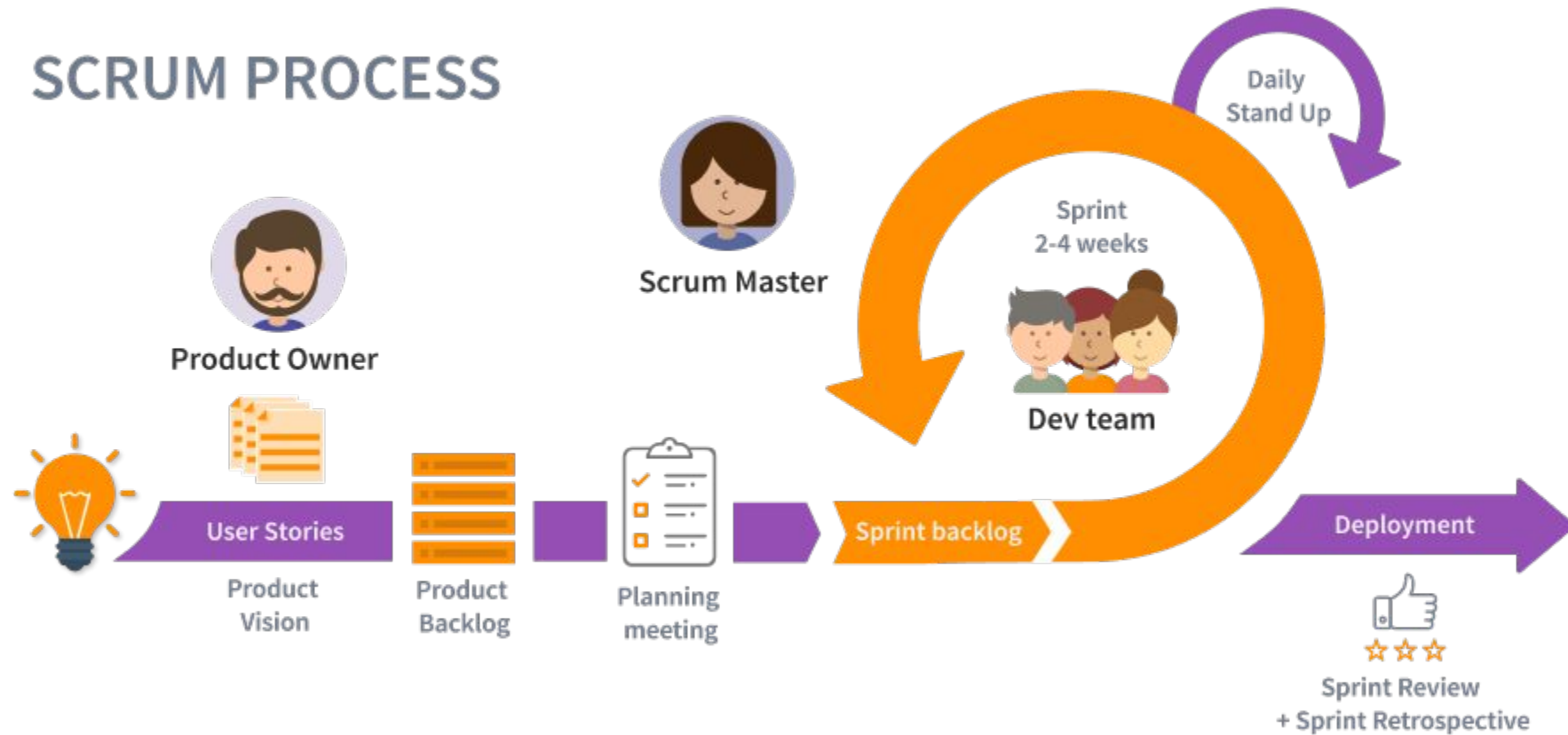
- Alto nível de stress nas equipas de desenvolvimento;
- Clientes podem criar ciclos de inoperabilidade;
- Difícil estabelecer um fim concreto do projeto;
- Degradação da estrutura do projeto a cada iteração.

Scrum

- Framework de gerenciamento de projetos;
- Baseado em desenvolvimento ágil;
- Atualmente utilizado em mais de 60% dos projetos ágeis;
- Não é um processo linear, consiste antes num conjunto de técnicas e processos;

SCRUM

SCRUM PROCESS



Características do SCRUM

- Framework de gerenciamento de projetos;
- Clientes tornam-se parte da equipe de desenvolvimento (os clientes devem estar genuinamente interessados na saída);
- Entregas frequentes e intermediárias de funcionalidades 100% desenvolvidas;
- Planos frequentes de mitigação de riscos desenvolvidos pela equipe;
- Discussões diárias de status com a equipe de desenvolvimento;

Características do SCRUM

- A discussão diária na qual cada membro da equipe de desenvolvimento responde às seguintes perguntas:
 - O que fiz desde ontem em direção a meta?
 - O que estou a pensar fazer até amanhã em direção a meta?
 - Existe algo que me impede de atingir a meta?
- Transparência no planeamento e desenvolvimento;

Características do SCRUM

- Reuniões frequentes com os *stakeholders* (partes interessadas no projeto) para monitorar o progresso;
- Problemas não são ignorados e ninguém é penalizado por reconhecer ou descrever qualquer problema não visto;
- Locais e horas de trabalho devem ser energizadas, no sentido de que "trabalhar horas extras" não necessariamente significa "produzir mais".

Sprint

- Um sprint é a unidade básica de desenvolvimento em Scrum. Sprints tendem a durar entre uma semana e um mês, e são um esforço dentro de uma faixa de tempo (ou seja, restrito a uma duração específica).
- A adoção de ciclos relativamente curtos permite entregas de partes dos sistemas, gerando valor para os clientes e permitindo uma avaliação dinâmica do trabalho.

Sprint

- Durante cada sprint, a equipe cria um incremento de produto potencialmente entregável (por exemplo, software funcional e testado). O conjunto de funcionalidades que entram em uma sprint vêm do Product Backlog, que é um conjunto de prioridades de requisitos de alto nível definidos pelo **Product Owner**.
- Quais itens do backlog que entram para a sprint são determinados durante a reunião de planeamento da sprint (Sprint Planning). Durante esta reunião, o **Product Owner** informa a equipe dos itens no backlog do produto que ele ou ela quer concluídos.

Sprint

- Durante cada sprint, a equipe cria um incremento de produto potencialmente entregável (por exemplo, software funcional e testado). O conjunto de funcionalidades que entram em uma sprint vêm do Product Backlog, que é um conjunto de prioridades de requisitos de alto nível definidos pelo **Product Owner**.
- Quais itens do backlog que entram para a sprint são determinados durante a reunião de planeamento da sprint (Sprint Planning). Durante esta reunião, o **Product Owner** informa a equipe dos itens no backlog do produto que ele ou ela quer concluídos.

Durante um sprint, ninguém está autorizado a alterar o backlog da sprint, o que significa que os requisitos são congelados para essa sprint.

Como implementar o Scrum?

- O Scrum pode ser implementado através de uma ampla gama de ferramentas.
- Muitas empresas utilizam ferramentas diversas para construir e manter artefatos como o backlog da sprint.
- Há também pacotes de software open-source e proprietários dedicados à gestão de produtos no âmbito do processo Scrum.
- Outras organizações implementam o Scrum sem o uso de quaisquer ferramentas de software, e mantêm seus artefatos na forma de cópias impressas, como papel, quadros e notas.

Papeis em Scrum

- O **Scrum Master**, que mantém os processos (normalmente no lugar de um gestor de projeto);
- O **Product Owner** (Dono do Produto), que representa os *stakeholders* e o negócio (ou seja, o cliente);
- A **equipa de desenvolvimento**, (ou DevTeam), um grupo multifuncional entre 3 a 9 pessoas e que fazem a análise, projeto, implementação, teste etc.

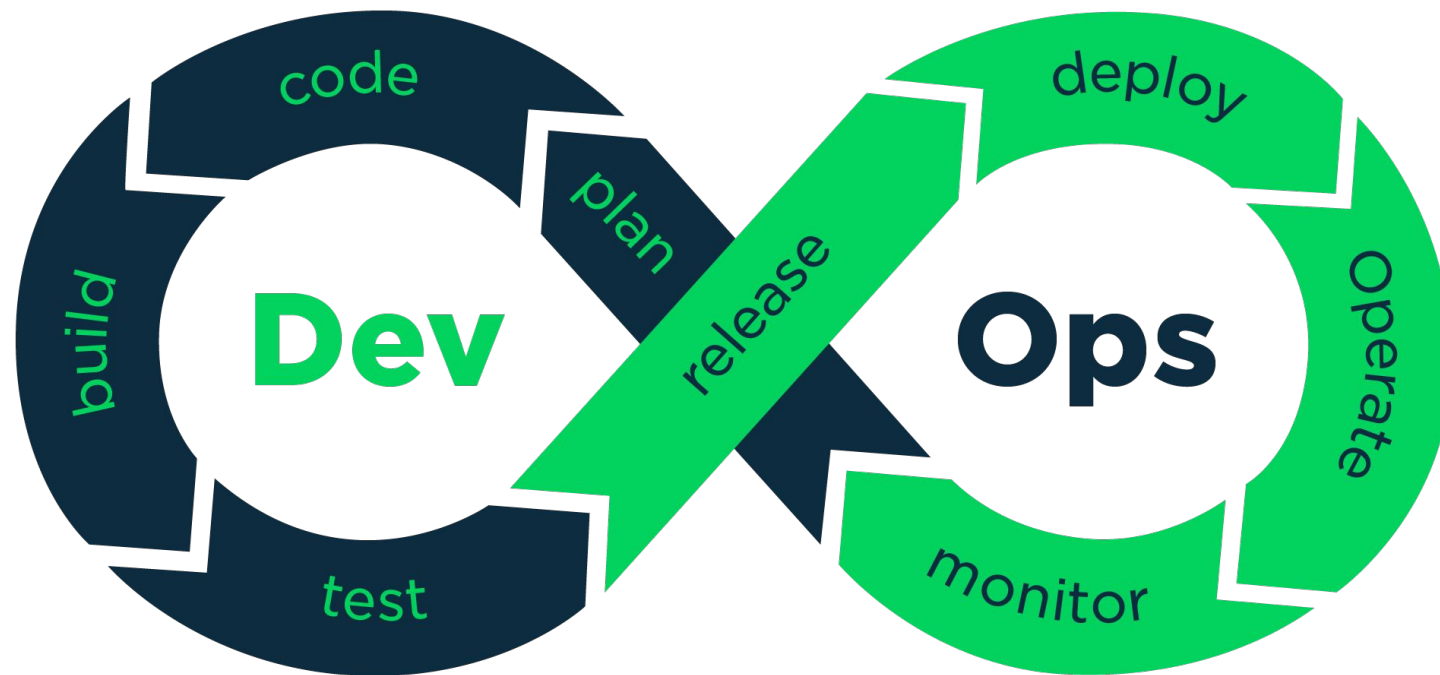
DevOps

Baseado na metodologia *Agile*, tem como objetivo misturar o trabalho de duas equipas - desenvolvimento (Dev) e operações (Ops).

É uma filosofia e um paradigma cultural nas equipas de desenvolvimento de software.

Toma proveito da automação e de ferramentas de engenharia de software para criar ambientes de desenvolvimento em que todos os membros da equipa partilham funções.

DevOps



CASE

CASE (**C**omputer-**A**ided **S**oftware **E**ngineering) - Conjunto de ferramentas de auxílio e automação de desenvolvimento de software:

- **Upper-CASE** - Ferramentas de alto nível, suporte inicial. Servem para criar ou editar documentos, diagramas e como forma de representação para o cliente (Visio, PowerPoint, Word, etc.)
- **Lower-CASE** - Ferramentas de desenvolvimento de software baixo nível ou de suporte direto a quem desenvolve. (IntelliJ IDEA, Visual Studio, Notepad++, etc)

Outras Ferramentas

- **Git**
- **StackOverflow**
- **Docker**
- **Heroku**
- **Atlassian Jira / JetBrains Space**

O futuro profissional começa aqui

iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA



emprego
digital



UPskill