

Processos de Desenvolvimento de Software

Modelo de Cascata (Waterfall)

Metodologias ágeis (Scrum, KanBan, XP)

Modelo DevOps

Carlos Castro

03/11/2025

Introdução

O desenvolvimento de software envolve diferentes metodologias que orientam a forma como as equipas planeiam, constroem e testam sistemas.

Cada modelo de processo apresenta fases e ferramentas próprias, adequadas a diferentes contextos de projeto. Este trabalho analisa três tipos de abordagens — **Cascata**, **Metodologias Ágeis** e **DevOps** — destacando as suas características, vantagens e limitações.

O objetivo é compreender a evolução desses modelos e identificar qual se adapta melhor a diferentes tipos de projeto e ambientes de trabalho.

Modelo Cascata (Waterfall)

Descrição geral

O modelo em cascata foi proposto por Winston W. Royce em 1970 e representa a abordagem mais tradicional de desenvolvimento de software. Baseia-se numa sequência linear de etapas, em que o avanço para a fase seguinte só ocorre após a conclusão da anterior.

É amplamente utilizado em projetos com requisitos estáveis e alta exigência de documentação.



Fases principais

- 1) **Recolha de requisitos** – definição detalhada das necessidades do cliente e especificações do sistema.
- 2) **Análise** – estudo de viabilidade, modelação de dados e definição funcional.
- 3) **Desenho (Design)** – planeamento da arquitetura, interfaces e estrutura interna.
- 4) **Implementação** – codificação e integração dos módulos definidos.
- 5) **Testes** – validação do sistema completo e verificação de conformidade com os requisitos.
- 6) **Manutenção** – correções, melhorias e adaptações após a entrega.

Esta metodologia utiliza forte documentação, diagramas UML, planeamento com cronogramas (Gantt), e ferramentas de gestão como **Microsoft Project** ou **Enterprise Architect**.



Modelo Cascata (Waterfall)

Vantagens

- Estrutura clara e previsível.
- Facilidade de controlo e gestão.
- Adequado a projetos com requisitos fixos e equipas pequenas.

Desvantagens

- Pouca flexibilidade a mudanças.
- Feedback do cliente apenas nas fases finais.
- Custos altos para correções futuras.

Exemplos de aplicação.

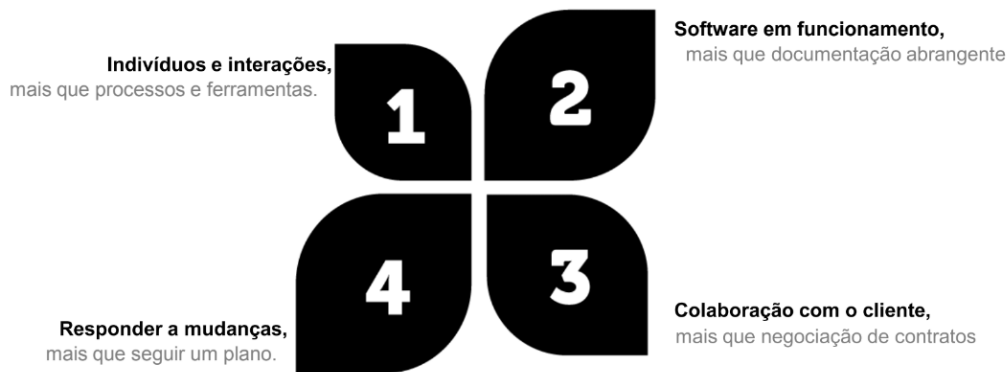
Adotado por organizações como a **NASA**, **Boeing** e entidades governamentais, especialmente em sistemas críticos onde a validação rigorosa é essencial, como softwares de controlo de voo e defesa.



Metodologias Ágeis

As metodologias ágeis surgiram no início dos anos 2000, formalizadas com o *Manifesto Ágil* em 2001. Propõem um processo de desenvolvimento iterativo e incremental, centrado na colaboração entre equipas, na entrega contínua de valor e na capacidade de adaptação a mudanças frequentes. Entre as abordagens mais conhecidas estão o Scrum e o Kanban.

MANIFESTO ÁGIL



Fases principais

- 1) **Planeamento** – definição dos objetivos e priorização das tarefas.
- 2) **Desenvolvimento iterativo** – implementação em ciclos curtos chamados *sprints*.
- 3) **Revisão** – demonstração dos resultados ao cliente e recolha de feedback.
- 4) **Retrospectiva** – análise interna da equipa para otimização do processo.

Esta metodologia utiliza quadros de tarefas (Kanban boards), reuniões diárias (*daily stand-up*), gestão de backlog e métricas de progresso (burndown charts). Ferramentas comuns incluem **Jira**, **Trello**, **Asana** e **ClickUp**.



Metodologias Ágeis

Vantagens

- Elevada flexibilidade e resposta rápida a mudanças.
- Entregas frequentes e foco no valor entregue ao cliente.
- Melhoria contínua e forte comunicação na equipa.

Desvantagens

- Menor formalidade na documentação.
- Pode gerar instabilidade se o cliente alterar frequentemente os requisitos.
- Requer equipas experientes e autogeridas.

Exemplos de aplicação.

Amplamente utilizado por empresas tecnológicas como Spotify, Google e Atlassian, em projetos de software que evoluem continuamente, como aplicações web, plataformas de streaming e serviços online.



DevOps

O DevOps é uma abordagem moderna que integra as equipas de desenvolvimento (Dev) e operações (Ops), promovendo colaboração, automação e entrega contínua de software. Surgiu por volta de 2009 como evolução natural das metodologias ágeis, com o objetivo de reduzir o tempo entre o desenvolvimento e a disponibilização de novas versões em produção.

Fases principais

- 1) **Planeamento** – definição dos objetivos, métricas e tarefas.
- 2) **Codificação** – desenvolvimento e versionamento do código.
- 3) **Integração e build** – compilação e testes automáticos.
- 4) **Entrega e implantação** – automação do *deploy*.
- 5) **Monitorização** – acompanhamento de desempenho e deteção de falhas.
- 6) **Feedback contínuo** – recolha de dados e melhoria do ciclo seguinte.

Baseia-se em automação e integração contínua. As ferramentas mais usadas incluem **GitHub Actions**, **Jenkins**, **Docker**, **Kubernetes**, **Terraform** e **Prometheus**. Valoriza cultura colaborativa, pipelines CI/CD e infraestrutura como código.



GitHub Actions

DevOps

Vantagens

- Entregas rápidas e frequentes.
- Redução de erros por automação.
- Integração contínua entre equipas e sistemas.
- Melhor fiabilidade e escalabilidade de aplicações.

Desvantagens

- Requer maturidade técnica e cultural.
- Maior complexidade inicial e custo de implementação.
- Necessidade de infraestrutura robusta e monitorização constante.

Exemplos de aplicação

Empresas como Netflix, Amazon e Microsoft adotam DevOps em larga escala, utilizando pipelines automatizados e monitorização em tempo real para garantir atualizações contínuas em serviços globais.



Comparação direta

	Waterfall (Cascata)	Metodologias ágeis	DevOps
Estrutura	Linear e sequencial	Iterativa	Contínua
Flexibilidade	Baixa	Alta	Muito alta
Entregas	Apenas no final	Frequentes	Contínuas
Feedback	Tardio	Rápido	Constante
Documentação	Extensa	Moderada	Automatizada
Ferramentas	MS Project, UML	Jira, Trello	Jenkins, Docker
Tipo de projeto	Requisitos fixos	Em evolução	Sistemas online
Exemplo	NASA, Boeing	Spotify, Google	Netflix, Amazon

Conclusão

Os três modelos mostram diferentes formas de criar software. O **Cascata** é o mais antigo e segue etapas fixas, bom para projetos estáveis.

O **Ágil** é mais flexível e permite corrigir e melhorar a cada fase. O **DevOps** junta desenvolvimento e operações, usando automação para lançar versões mais rápidas. Cada modelo serve melhor em situações diferentes, e o importante é escolher o que se adapta ao tipo de projeto e à equipa.

Bibliografia

Microsoft Azure. **O que é DevOps**. Disponível em: <https://azure.microsoft.com/pt-pt/resources/cloud-computing-dictionary/what-is-devops>

Atlassian. **DevOps: Conceitos e Práticas**. Disponível em: <https://www.atlassian.com/br/devops>

Tera Blog. **Modelo Cascata no Desenvolvimento Web**. Disponível em: <https://blog.somostera.com/desenvolvimento-web/modelo-cascata>

Robson Camargo. **Método Cascata: O Que É e Como Funciona**. Disponível em: <https://robsoncamargo.com.br/blog/metodo-cascata>

Mosaico.Gov. **Metodologias Ágeis na Administração Pública**. Disponível em: <https://mosaico.gov.pt/areas-tecnicas/metodologias-ageis>