

Conhecendo os fluxos

1. Fluxo de Processo Linear

Também conhecido como **Modelo Cascata (Waterfall)**, segue uma sequência rígida de fases:

- Requisitos → Projeto → Implementação → Testes → Manutenção.
Cada fase só começa quando a anterior termina.

☑ **Vantagens:** Simples, bem estruturado, fácil de gerenciar.

✗ **Desvantagens:** Pouca flexibilidade para mudanças; problemas descobertos tardiamente.

2. Fluxo de Processo Iterativo

Nesse modelo, o software é desenvolvido em ciclos (iterações). Cada ciclo passa por todas as fases do processo, mas foca em partes do sistema.

☑ **Vantagens:** Permite feedback constante e melhorias contínuas.

✗ **Desvantagens:** Requer mais planejamento e envolvimento contínuo do cliente.

3. Fluxo de Processo Evolucionário

É uma variação do iterativo, mas com foco maior em **prototipagem rápida e evolução contínua** do sistema com base em feedback dos usuários.

☑ **Vantagens:** Ideal para projetos onde os requisitos não são totalmente definidos desde o início.

✗ **Desvantagens:** Pode gerar código desorganizado se não houver disciplina.

4. Fluxo de Processo Paralelo

Permite que **diferentes partes do sistema sejam desenvolvidas simultaneamente** por equipes distintas.

☑ **Vantagens:** Aumenta a produtividade e aproveita equipes maiores.

✗ **Desvantagens:** Exige integração e comunicação eficiente para evitar conflitos entre os módulos.

5. Modelo de Ciclo de Vida Iterativo e Incremental

Combina o desenvolvimento **por partes (incremental)** com o **refinamento contínuo (iterativo)**.

- A cada iteração, um **incremento funcional** do sistema é entregue.

☑ **Vantagens:** Entregas frequentes e evolutivas, fácil adaptação a mudanças.

✗ Desvantagens: Pode ser complexo de gerenciar em projetos muito grandes sem ferramentas adequadas

Quando saber qual adotar?

☒ 1. Clareza dos Requisitos

- Requisitos bem definidos e estáveis?
→ **Fluxo Linear** funciona bem.
- Requisitos incertos ou sujeitos a mudanças?
→ Prefira **Iterativo**, **Evolucionário** ou **Iterativo e Incremental**.

☒ 2. Envolvimento do Cliente

- Cliente disponível para acompanhar e validar?
→ **Iterativo**, **Evolucionário** ou **Incremental** são ideais.
- Cliente pouco disponível?
→ **Linear** ou **Paralelo**, com planejamento e entregas mais fechadas.

☒ 3. Tamanho e Complexidade do Projeto

- Projetos pequenos e simples?
→ **Linear** pode ser suficiente.
- Projetos médios ou grandes, com riscos?
→ **Iterativo e Incremental** é mais seguro.

☒ 4. Prazo de Entrega

- Entrega única ao final do projeto?
→ **Linear** ou **Paralelo**.
- Necessário entregar partes funcionais ao longo do tempo?
→ **Incremental** ou **Evolucionário**.

☒ 5. Equipe e Recursos

- Equipe pequena e com pouca experiência?
→ Um modelo mais simples como o **Linear** pode ser mais fácil de gerenciar.
- Equipe experiente e multidisciplinar?
→ Pode lidar bem com fluxos **Iterativos**, **Evolutivos** ou **Paralelos**.

☒ 6. Riscos do Projeto

- Alto risco técnico ou de requisitos?
→ Use **Iterativo**, **Evolucionário** ou **Prototipagem rápida**.

- **Baixo risco e escopo bem definido?**
→ **Linear** ou **Paralelo**.

🔗 Resumo prático

Cenário	Melhor fluxo sugerido
Requisitos fixos	Linear
Requisitos mudam com frequência	Iterativo ou Evolucionário
Entregas parciais são úteis	Iterativo e Incremental
Equipe grande e modular	Paralelo
Cliente acompanha de perto	Iterativo ou Evolucionário
Projeto arriscado	Iterativo ou com prototipação evolutiva

Possíveis cenários no mercado

◇ 1. Requisitos bem definidos → Fluxo Linear

Exemplo:

Uma empresa contratou o desenvolvimento de um sistema de folha de pagamento.

- Todos os cálculos, regras e relatórios já estão bem documentados.
 - O cliente não deseja mudanças após o início do projeto.
- ☒ **Modelo ideal:** Fluxo Linear (Cascata) – porque cada fase pode ser feita de forma sequencial sem surpresas.

◇ 2. Requisitos mudam com frequência → Fluxo Iterativo ou Evolucionário

Exemplo:

Startup quer desenvolver um aplicativo de rede social.

- Os fundadores ainda estão validando o modelo de negócio.
 - Funcionalidades mudam a cada semana com base no feedback dos primeiros usuários.
- ☒ **Modelo ideal:** Fluxo Iterativo ou Evolucionário – permite adaptar rapidamente o projeto.

◇ 3. Entregas parciais são necessárias → Iterativo e Incremental

Exemplo:

Sistema de gestão escolar para ser implantado por módulos:

- Primeiro o controle de alunos, depois notas, depois boletins e por fim, financeiro.

☒ **Modelo ideal:** Iterativo e Incremental – entrega cada módulo de forma funcional, com feedback.

◇ **4. Equipe grande e com papéis definidos → Fluxo Paralelo**

Exemplo:

Empresa de TI com 3 equipes trabalhando no mesmo sistema:

- Uma equipe cuida do backend, outra do frontend e outra da API.
☒ **Modelo ideal:** Fluxo Paralelo – cada time trabalha em paralelo em partes diferentes, com integração no final.

◇ **5. Cliente participa ativamente → Iterativo ou Evolucionário**

Exemplo:

Cliente do setor bancário quer um sistema de atendimento digital.

- Participa de reuniões semanais, testa funcionalidades e dá feedback constante.
☒ **Modelo ideal:** Iterativo ou Evolucionário – adapta rapidamente com base nas opiniões do cliente.

◇ **6. Projeto de alto risco técnico → Iterativo com prototipagem**

Exemplo:

Desenvolvimento de um sistema com IA para análise de crédito.

- Não se sabe se os algoritmos vão funcionar bem com os dados reais.
- Testes e validações constantes são necessários.
☒ **Modelo ideal:** Iterativo com protótipos – possibilita validar ideias e mudar antes do produto final.

◇ **7. Projeto pequeno e com pouco recurso → Linear**

Exemplo:

Criação de um site institucional para uma empresa local.

- Escopo fixo, design pronto, conteúdo fornecido.
☒ **Modelo ideal:** Fluxo Linear – simples, rápido e direto.

Tornar cada cenário um caso de sucesso

☑ 1. Fluxo Linear — Como ter sucesso

📌 **Cenário:** Sistema de folha de pagamento com requisitos fixos

✂ **Boas práticas:**

- Faça um levantamento de requisitos bem detalhado e validado com o cliente.
- Faça revisões formais após cada fase (documentação, design, etc.).
- Use cronogramas com marcos bem definidos. ☑ **Chave do sucesso:** Documentação sólida e controle rígido de escopo.

☑ 2. Fluxo Iterativo — Como ter sucesso

📌 **Cenário:** Startup com app em constante mudança

✂ **Boas práticas:**

- Realize iterações curtas (sprints) com entregas funcionais.
- Colete feedback real dos usuários a cada versão.
- Faça retrospectivas para melhorar o processo continuamente. ☑ **Chave do sucesso:** Feedback rápido + adaptação contínua + priorização de funcionalidades.

☑ 3. Fluxo Evolucionário — Como ter sucesso

📌 **Cenário:** Projeto com protótipos e mudanças rápidas

✂ **Boas práticas:**

- Desenvolva protótipos visuais e/ou funcionais rapidamente.
- Envolve o cliente desde o início em testes e revisões.
- Mantenha versões simples e em evolução constante. ☑ **Chave do sucesso:** Proximidade com o cliente + prototipagem iterativa.

☑ 4. Fluxo Paralelo — Como ter sucesso

📌 **Cenário:** Projeto grande com equipes separadas

✂ **Boas práticas:**

- Faça planejamento de dependências entre os módulos (quem precisa de quê e quando).
- Utilize ferramentas de integração contínua (CI/CD).
- Estabeleça contratos de interface entre os módulos (API, dados, etc.). ☑ **Chave do sucesso:** Comunicação entre times + integração constante.

☑ 5. Fluxo Iterativo e Incremental — Como ter sucesso

📌 **Cenário:** Sistema de gestão escolar entregue por módulos

✂ **Boas práticas:**

- Planeje os incrementos de forma que cada um tenha valor de negócio.
- Faça testes reais com os usuários finais de cada módulo.
- Reavalie o backlog entre as entregas para ajustar prioridades. ☑ **Chave do sucesso:** Entregas frequentes com valor + feedback real.

☑ 6. Prototipagem com risco técnico — Como ter sucesso

📌 **Cenário:** Projeto com IA e incertezas técnicas

✂ **Boas práticas:**

- Comece com protótipos de baixa fidelidade para validar hipóteses.
- Faça POCs (Provas de Conceito) para testar viabilidade técnica.
- Documente aprendizados técnicos para as próximas iterações. ☑ **Chave do sucesso:** Teste rápido de ideias + validação técnica antecipada.

☑ 7. Projeto pequeno e direto — Como ter sucesso

📌 **Cenário:** Site institucional simples

✂ **Boas práticas:**

- Entenda claramente o que o cliente precisa e defina bem o escopo.
- Evite mudanças de última hora com contratos ou termos de aceite.
- Use templates e boas práticas de desenvolvimento para agilizar. ☑ **Chave do sucesso:** Simplicidade + eficiência + escopo controlado.

Entender quem são os envolvidos (atores) e qual o papel de cada um em cada tipo de fluxo de processo

☑ 1. Fluxo Linear

📌 **Cenário:** Projeto com requisitos fixos

Ator	Papel
Analista de Requisitos	Levanta e documenta os requisitos no início.
Gerente de Projeto	Planeja e controla as fases do projeto com rigidez.
Desenvolvedor	Implementa com base no design definido, sem mudanças.

Testador (QA)	Executa testes no final, com base em requisitos documentados.
Cliente	Participa no início (requisitos) e no fim (validação e entrega final).

📌 **Ênfase:** Rigor no planejamento e documentação.

☑ 2. Fluxo Iterativo

🔗 **Cenário:** Projeto com mudanças frequentes

Ator	Papel
Product Owner (PO)	Define prioridades, ajusta backlog a cada iteração.
Scrum Master / Facilitador	Garante a fluidez do processo iterativo e remove impedimentos.
Desenvolvedores	Entregam incrementos funcionais em cada sprint.
Cliente/Usuário Final	Fornece feedback constante a cada entrega.
Testador Integrado (QA)	Valida qualidade durante e após cada iteração.

📌 **Ênfase:** Adaptação, entregas rápidas, colaboração contínua.

☑ 3. Fluxo Evolucionário

🔗 **Cenário:** Protótipos e evolução contínua

Ator	Papel
UX/UI Designer	Cria protótipos e mockups para testes com usuários.
Cliente/Stakeholder	Valida os protótipos e sugere melhorias frequentes.
Desenvolvedores	Convertem protótipos em software funcional aos poucos.
Analista de Negócios	Traduz feedbacks em melhorias incrementais.
Facilitador de Processo	Garante ciclos curtos e contínuos de evolução.


📌 **Ênfase:** Validação contínua de ideias, evolução baseada em feedback.

☑ 4. Fluxo Paralelo


🔗 **Cenário:** Desenvolvimento simultâneo em times

Ator	Papel
Arquiteto de Software	Define a estrutura geral e as interfaces entre os módulos.
Líder Técnico de Equipe	Coordena o desenvolvimento paralelo de um módulo específico.

Desenvolvedores	Trabalham em partes independentes do sistema.
Especialista DevOps/CI	Garante integração contínua entre módulos.
Gerente de Integração	Planeja os pontos de união entre os times.

 **Ênfase:** Coordenação entre equipes, integração constante.

5. Iterativo e Incremental

 **Cenário:** Sistema modular entregue por partes

Ator	Papel
Product Owner	Define a ordem dos incrementos com base em valor para o negócio.
Scrum Master / Facilitador	Organiza as iterações e garante entregas parciais.
Desenvolvedores	Constróem e entregam partes completas do sistema.
Testador / QA Integrado	Valida cada incremento isoladamente e o sistema como um todo.
Usuário Final	Usa as partes entregues e dá feedback.

 **Ênfase:** Valor por entrega + adaptação contínua.


6. Prototipagem com risco técnico

 **Cenário:** Projeto com incertezas tecnológicas (ex: IA, inovação)

Ator	Papel
Pesquisador Técnico / Cientista de Dados	Explora e valida tecnologias ou algoritmos.
Desenvolvedor POC	Cria provas de conceito rapidamente.
Gerente de Riscos	Identifica pontos críticos e acompanha decisões técnicas.
Stakeholders Técnicos	Validam se os resultados atendem aos objetivos do negócio.
Cliente/Usuário Piloto	Testa versões experimentais e fornece insights.


 **Ênfase:** Validação técnica, exploração rápida, mitigação de riscos.

7. Projeto pequeno e direto

 **Cenário:** Site institucional simples

Ator	Papel
Cliente	Define o conteúdo, público e layout esperado.

Desenvolvedor Web	Constrói o site com base no briefing inicial.
Designer Gráfico	Cria a identidade visual e layouts.
Testador (ou próprio dev)	Realiza testes básicos de funcionalidade e responsividade.

 **Ênfase:** Clareza, rapidez e simplicidade.