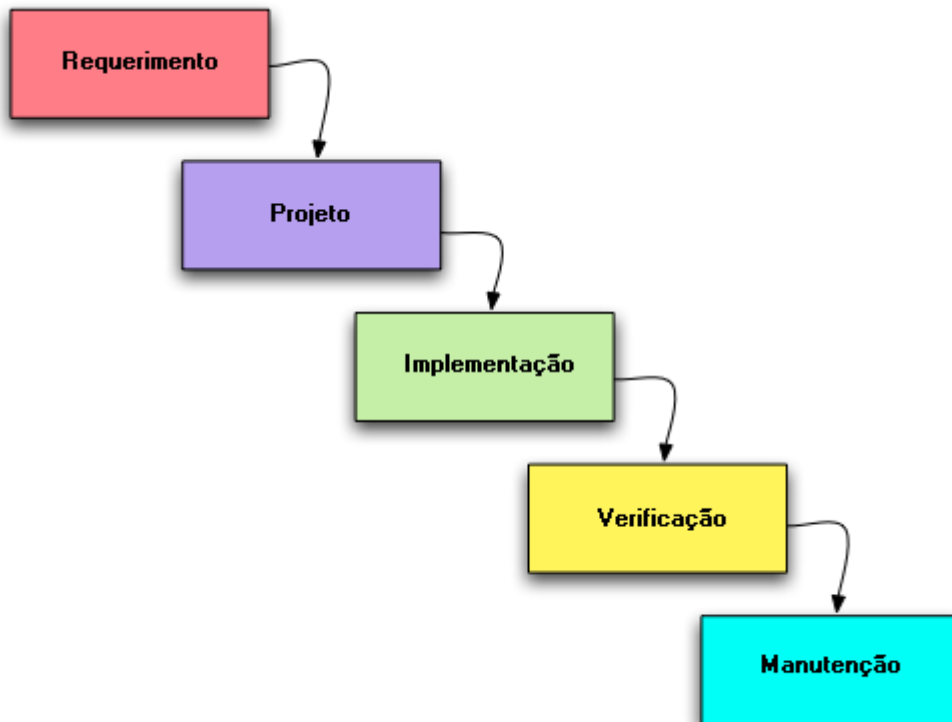


ENGENHARIA DE SOFTWARE

Metodologias Tradicionais

Uma **metodologia de desenvolvimento** é como uma receita de bolo para criar software. Ela define passos, regras e práticas a seguir. As tradicionais são mais antigas e geralmente mais rígidas.

1. Cascata (Waterfall)



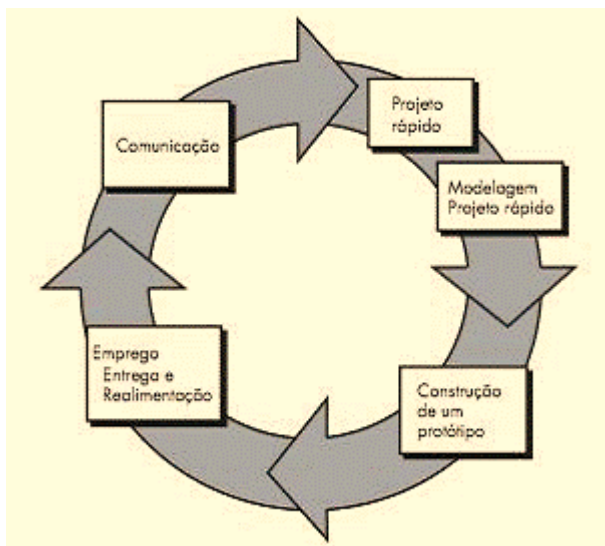
Imagine uma cachoeira: a água só desce. No modelo Cascata, você só passa para a próxima fase depois de terminar completamente a anterior.

- **Características:** Sequencial, linear, muita documentação, planejamento detalhado no início.
- **Fases Típicas:**
 1. **Levantamento de Requisitos:** Entender o quê o cliente precisa. O foco é compreender o problema e as funcionalidades. (Não inclui engenharia, projeto ou prototipação aqui.)

2. **Análise/Projeto (Design):** Pensar em como o software será construído (arquitetura, telas, banco de dados).
3. **Implementação (Codificação):** Escrever o código do programa.
4. **Verificação (Testes):** Verificar se tudo funciona como planejado e corrigir erros.
5. **Manutenção:** Corrigir problemas que aparecem depois ou adicionar melhorias.

- **Bom para:** Projetos onde os requisitos são muito claros e não mudam no meio do caminho.
- **Ruim para:** Projetos onde as necessidades podem mudar, pois é muito inflexível. O cliente só vê o resultado final muito tarde.

2. Prototipação (Prototyping)



Aqui, a ideia é criar uma versão simplificada (um protótipo) rapidamente para mostrar ao cliente e pegar feedback.

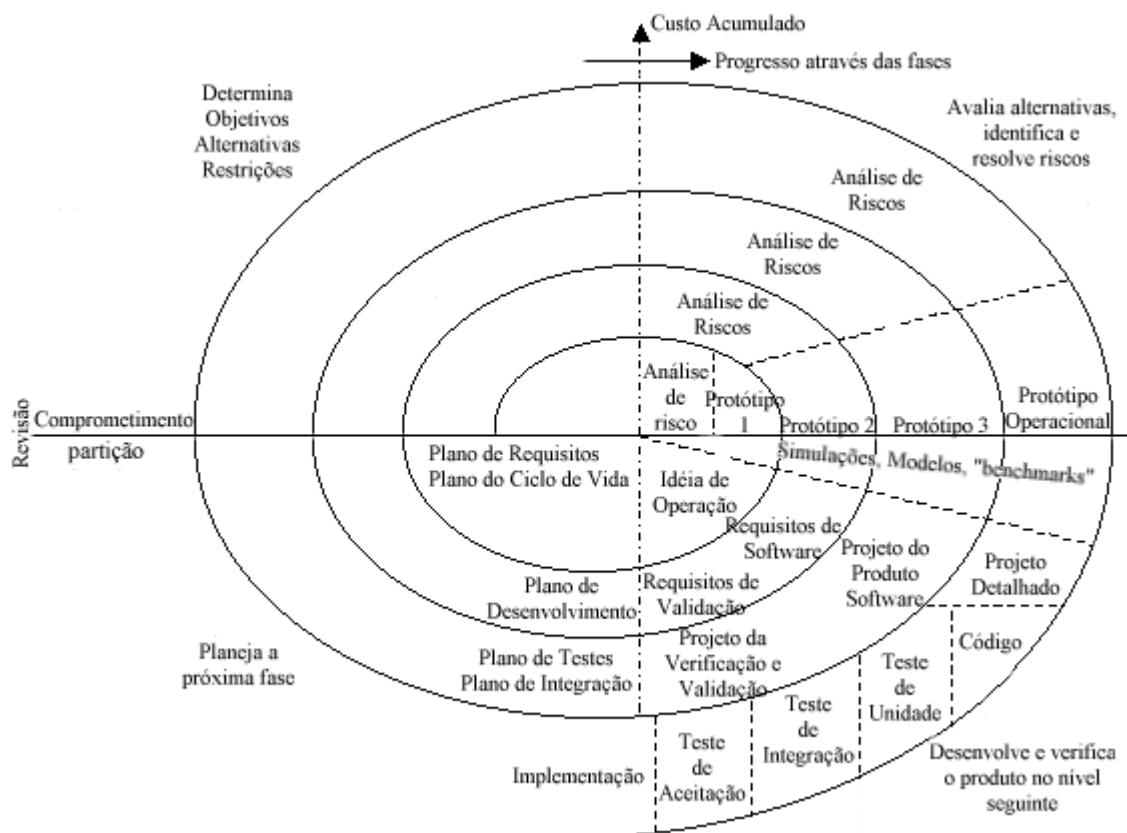
- **Características:** Foco em experimentação, feedback rápido, ideal para requisitos incertos.
- **Fases (Ciclo):**
 1. **Coleta de Requisitos Inicial:** Entender o básico.
 2. **Projeto Rápido:** Desenhar o protótipo rapidamente.
 3. **Construção do Protótipo:** Criar a versão simplificada.
 4. **Avaliação:** Mostrar ao cliente e coletar opiniões.

5. **Refinamento:** Ajustar o protótipo com base no feedback. (Repete 2 a 5 até o protótipo ser aprovado).

6. **Engenharia do Produto:** Construir o software final, possivelmente aproveitando partes do protótipo aprovado.

- **Bom para:** Esclarecer requisitos, melhorar a comunicação com o cliente.
- **Ruim para:** O cliente pode achar que o protótipo já é o produto final; gerenciar o processo pode ser mais difícil.

3. Espiral (Spiral)



Combina a ideia de fases sequenciais (Cascata) com a criação de protótipos e adiciona um foco forte em **Análise de Riscos** a cada volta (ciclo).

- **Características:** Iterativo (ciclos), focado em riscos, flexível, bom para projetos grandes e complexos.
- **Fases (em cada ciclo/quadrante):**
 1. **Determinação de Objetivos:** Definir o que fazer neste ciclo, alternativas.
 2. **Análise de Riscos:** Identificar e planejar como lidar com possíveis problemas. (O desenvolvimento/codificação não acontece aqui).

3. **Desenvolvimento e Validação:** Construir e testar a parte do software correspondente a este ciclo (pode envolver criar um protótipo ou parte do sistema final).
4. **Planejamento da Próxima Fase:** Revisar o que foi feito e planejar a próxima volta da espiral, incluindo **Avaliação do Cliente**.

- **Bom para:** Gerenciar riscos em projetos complexos.
- **Ruim para:** Pode ser complexo demais para projetos simples; a análise de riscos exige experiência. As fases **não** são idênticas às do RUP.

Comparativo e Limitações das Tradicionais

Característica	Cascata	Prototipação	Espiral
Flexibilidade a Mudanças	Baixa	Média	Alta
Envolvimento do Cliente	Baixo	Alto	Médio
Gerenciamento de Riscos	Baixo	Médio	Alto
Ideal para Requisitos	Estáveis	Incertos	Complexos/Incerto
Primeira Entrega Visível	Tarde	Cedo	Médio
Documentação	Muita	Pouca/Média	Média/Muita

Ponto em comum: Metodologias tradicionais (especialmente Cascata) podem ser muito rígidas. Elas funcionam melhor quando os requisitos são bem definidos no início e não mudam muito, facilitando o planejamento e gerenciamento. Elas **usam** documentação, planejamento e processos, ao contrário do que algumas afirmações podem sugerir.