

Teoria Geral de Sistemas

Aula 2 – Ciclos de Vida do Software

Prof. Ana Carolina Sokolonski

Bacharelado em Sistemas de Informação

Instituto Federal da Bahia – Campus Feira de Santana

carolsoko@ifba.edu.br

2026



Processo de Desenvolvimento

Cascata e Modelo em V

Incremental, Evolutivo e RAD

Prototipagem e Espiral

Comparação

Exercício

Referências

Processo de Desenvolvimento de Software

Ciclo de vida e artefatos

▪ Definição (NBR ISO/IEC 12207:1998)

“Estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema, desde a definição de seus requisitos até o término de seu uso.”

É a **primeira escolha** no processo de desenvolvimento. Define quando e como o cliente receberá a primeira versão operacional do sistema.

▪ Processo de Software

Conjunto de atividades agrupadas em fases: requisitos, análise, projeto, desenvolvimento, teste e implantação. Em cada fase são definidas funções da equipe e **artefatos** resultantes.

▪ Diferença dos modelos

- Ordem em que as fases ocorrem
- Tempo e ênfase dados a cada fase
- Atividades presentes
- Produtos entregues

▪ Comportamentos dos ciclos

Sequencial	fases em ordem fixa
Incremental	divisão de escopo
Iterativo	retroalimentação entre fases
Evolutivo	software aprimorado a cada ciclo

▪ Não existe ciclo ideal

Perfil do cliente, tempo, custo, equipe e ambiente operacional influenciam diretamente na escolha. Na prática, observam-se múltiplos ciclos no mesmo processo.

▪ Primeira versão ao cliente

Cascata / V	Apenas no fim do ciclo
Incremental/RAD	Protótipos operacionais
Evolutivo	Versões sucessivas
Prototipagem	Protótipos não-operacionais
Espiral	Protótipos (op. ou não)

▪ Grau de risco

Quanto mais tarde o cliente vê o sistema, **maior o risco** de insatisfação. Modelos iterativos e evolutivos reduzem o risco de feedback contínuo.



Cascata e Modelo em V

Sequenciais, documentados, baixo feedback

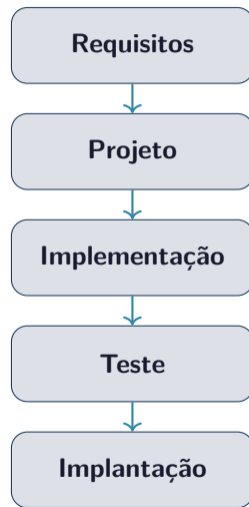
▪ Cascata (Royce, 1970)

O mais antigo. Cada fase **só começa quando a anterior termina**. A saída de cada fase é a entrada da seguinte.

Fases: Requisitos → Projeto → Implementação → Teste → Implantação → Manutenção

Vantagens: simples, fácil de entender, estimativa de custos no início, documentação sólida.

Problemas: cliente só vê o software no fim; **não prevê revisão de fases**; pressupõe uma realidade estática. Alto risco em sistemas complexos.



▪ Modelo em V (Alemanha, 1992)

Adaptação do Cascata em forma de V. Lado esquerdo: análise e projeto. Vértice: codificação. Lado direito: testes e implantação.

Característica principal: cada fase do lado esquerdo gera um **plano de teste** a ser executado no lado direito.

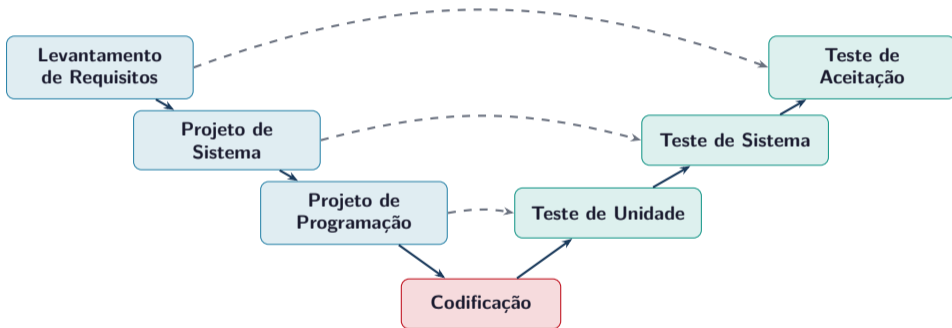
Teste de unidade ↔ Projeto de programa

Teste de sistema ↔ Projeto de sistema

Teste de aceitação ↔ Requisitos

Vantagem: menos risco que Cascata puro, pois testes são planejados antecipadamente.

Ciclo de Vida em Cascata e Modelo em V



Incremental, Evolutivo e RAD

Modelos com entregas parciais

▪ Incremental

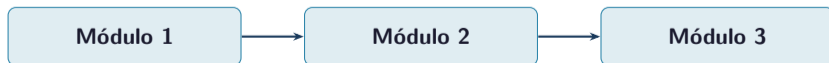
Requisitos agrupados em módulos por prioridade de negócio. Cada módulo passa pelo Cascata e é entregue como **software operacional**.

Cliente usa o sistema antes do fim. Feedback parcial. Exige atenção à integração entre os módulos.

Não exige equipe grande; sem paralelismo.

Risco: partes comuns mal desenvolvidas no primeiro módulo comprometem todo o sistema.

Modelo Incremental



▪ Evolutivo

Requisitos não totalmente conhecidos. A cada versão entregue, o cliente usa no ambiente real e retorna com **feedback** para a próxima versão.

Ótimo para sistemas novos (sem legado). Alta participação do cliente.

Problemas: pouca documentação, risco de nunca terminar, mudanças radicais entre versões, escopo indefinido. Exige alto gerenciamento.

Modelo Evolutivo



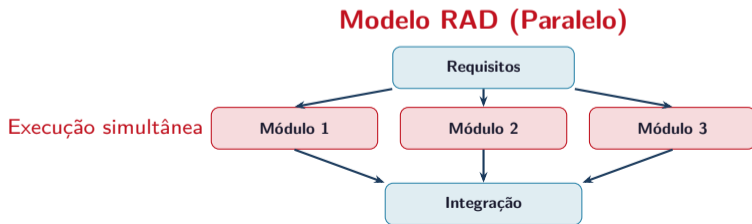
▪ RAD (Martin, 1991)

Evolutivo do Incremental com forte paralelismo. Duração típica: **60 a 90 dias**. Módulos desenvolvidos por equipes diferentes simultaneamente.

Ideal para lançar soluções pioneiras. Requisitos com escopo restrito.

Exige: profissionais experientes, empresa pouco burocrática, biblioteca de componentes reutilizáveis, ferramentas CASE maduras.

Risco: baixo desempenho e análise de risco superficial. Preferível para sistemas de pequeno porte.



Prototipagem e Espiral

Feedback intenso e gestão de riscos

▪ Prototipagem

Ideal quando os requisitos são complexos, confusos ou parcialmente desconhecidos. Cliente experimenta na prática antes do sistema final. **Tipos:**

Operacional: pode ser usado em produção.

Nao-operacional: apenas para comunicacao (telas, relatorios).

Protótipos podem ser descartados ou evoluídos.

Modelo de Prototipagem



▪ Espiral (Boehm, 1988)

Cíclico e guiado por **risco**. Ideal para sistemas complexos em que falhas não são toleráveis. A cada volta (iteração), quatro quadrantes:

1. Definição de Objetivos — levantar alternativas e plano gerencial
2. Análise de Riscos — avaliar riscos com protótipos
3. Desenvolvimento e Validação — escolhe ciclo adequado (cascata, iterativo...)
4. Planejamento da Próxima Fase — revisão e decisão

Pode incluir mais fases ou ficar adormecido até nova alteração. Difícil de vender ao cliente; exige alto nível de gerenciamento.



Comparação dos Ciclos de Vida

Ciclo	Requisitos	Primeira Versão	Feedback	Porte
Cascata	Bem conhecidos	Fim do ciclo	Baixo	Simple
Em V	Bem conhecidos	Fim do ciclo	Médio	Simple
Incremental	Módulos priorizados	Protótipos operacionais	Médio	Médio
Evolutivo	Pouco conhecidos	Protótipos operacionais	Alto	Médio
RAD	Escopo restrito	Protótipos operacionais	Alto	Pequeno
Prototipagem	Abstratos	Protótipos nao-op.	Alto	Médio
Espiral	Evoluídos	Protótipos (op. ou nao)	Alto	Complexo

Quanto mais tarde o cliente vê o sistema, maior o risco.

Feedback contínuo reduz retrabalho.

▪ Cenário

O sr. Carlos é dono de uma pequena rede de supermercados. Há 10 anos iniciou com um único armazém e ainda controla tudo manualmente. Decidiu informatizar os processos administrativos (folha de pagamento, contas a pagar/receber, contabilidade) contratando a empresa *Soft & Hard*.

O sr. Carlos não tem conhecimentos em informática e possui apenas uma **vaga ideia** de como o computador pode ajudá-lo e quais informações pode extrair para gerenciar o negócio.

▪ Questão

Se você fosse o **analista de sistemas** responsável por este projeto, qual o ciclo de vida você adotaria? **Justifique** com base nas características do cliente e nos conceitos estudados.




▪ Roteiro de análise

Considere:

- O cliente conhece bem seus requisitos?
- Há risco de mudança de requisitos ao longo do projeto?
- O cliente consegue participar ativamente?
- É necessário entregar algo operacional rapidamente?
- O sistema é simples, médio ou complexo?

▪ Dica

Relacione as características do sr. Carlos com as colunas **Requisitos**, **Feedback** e **Porte** da tabela de comparação.

-  SOMMERVILLE, I. *Engenharia de Software*. 10. ed. Pearson, 2016.
-  PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software: uma abordagem profissional*. 8. ed. McGraw-Hill, 2016.
-  ABNT NBR ISO/IEC 12207:1998. *Tecnologia de Informacao — Processos de Ciclo de Vida de Software*. Associação Brasileira de Normas Técnicas, 1998.