



INSTITUTO FEDERAL
CATARINENSE
Campus Camboriú

Curso: Bacharelado em Sistemas de Informação
Disciplina: Engenharia de Software I
Professor: Joice Seleme Mota

Atividade de Aprendizagem a Distância 1 (AD1)

Jonathan Bryan Kanssaw Gomez

1) O modelo cascata é ideal para softwares com requisitos bem definidos e poucas mudanças. Três exemplos são:

- **Sistema de Controle de Estoque** – Requisitos fixos, sem necessidade de atualizações frequentes.
- **Software de Folha de Pagamento** – Segue regras fiscais estáveis, exigindo precisão e planejamento.
- **Sistema de Gestão Escolar** – Funcionalidades claras, como cadastro de alunos e notas, sem grandes mudanças.

Esse modelo garante um desenvolvimento organizado e estruturado.

2) O modelo incremental é ideal para softwares que precisam ser desenvolvidos por etapas, permitindo ajustes e melhorias ao longo do tempo. Três exemplos são:

- **Aplicativo de Banco Digital** – Começa com funções básicas, como consulta de saldo, e depois adiciona pagamentos, transferências e investimentos.
- **Plataforma de Ensino Online** – Inicialmente lança cursos simples e, com o tempo, inclui videoaulas, fóruns e certificados.
- **Sistema de E-commerce** – Inicia com um catálogo de produtos e carrinho de compras, depois recebe melhorias como recomendações e suporte a múltiplos pagamentos.

Esse modelo permite entregas rápidas e evolução contínua do software.

3) O modelo SCRUM é ideal para softwares que precisam de entregas rápidas e constantes melhorias, com colaboração contínua da equipe. Três exemplos são:

- **Aplicativo de Rede Social** – Funcionalidades como postagens e curtidas são entregues primeiro, depois novas interações e melhorias são adicionadas com base no feedback dos usuários.
- **Sistema de Gestão Empresarial (ERP)** – Começa com módulos básicos, como financeiro e estoque, e novos módulos são desenvolvidos conforme as necessidades da empresa.
- **Plataforma de Streaming** – Lança primeiro com catálogo limitado e reprodução básica, depois recebe melhorias como sugestões personalizadas e novos formatos de mídia.

SCRUM permite flexibilidade e adaptação conforme as demandas do projeto evoluem.

- 4) Sim, é possível combinar diferentes modelos de desenvolvimento de software. Isso acontece quando um único modelo não atende bem todas as necessidades do projeto.

Exemplo: Um sistema bancário pode usar o **modelo cascata** para partes mais críticas, como segurança e transações, porque precisa ser bem planejado antes de ser usado. Já para o aplicativo do banco, pode ser usado **SCRUM**, pois permite atualizações frequentes e melhorias com base no feedback dos clientes.

Isso ajuda a manter o sistema seguro, mas também flexível para mudanças.

- 5) O modelo de processo espiral combina elementos do **modelo cascata** e do **modelo evolucionário**, permitindo um desenvolvimento mais flexível e seguro.

No **modelo cascata**, o desenvolvimento segue etapas sequenciais (requisitos, design, implementação, testes e manutenção). No **modelo evolucionário**, o software é construído em versões sucessivas, evoluindo conforme o feedback dos usuários.

O **modelo espiral** acomoda os dois ao dividir o desenvolvimento em ciclos (ou iterações). Cada ciclo segue a estrutura do **modelo cascata**, com planejamento, análise de riscos, desenvolvimento e avaliação. Mas, ao final de cada iteração, é possível revisar e melhorar o software, como no **modelo evolucionário**. Isso permite refinamento contínuo sem perder a organização do processo sequencial.

- 6) **Não**, uma organização não precisa usar um único modelo de processo.

Prós:

- **Padronização e facilidade de gestão.**
- Funciona bem para projetos sem mudanças frequentes.

Contras:

- Falta de **flexibilidade** para diferentes tipos de projeto.
- Pode ser **ineficiente** em alguns casos, forçando processos desnecessários.

Então, a escolha depende do tipo de projeto. Às vezes, é melhor usar modelos diferentes.

- 7) Modelos tradicionais, como o **Cascata**, seguem uma sequência fixa de etapas: planejamento, análise, design, implementação, testes e manutenção. São adequados quando os requisitos são bem definidos e estáveis.

Já os **Modelos Ágeis**, como o **Scrum**, enfatizam ciclos curtos de desenvolvimento (iterações), permitindo flexibilidade e adaptação contínua às mudanças. Focam na colaboração constante com o cliente e entregas frequentes de incrementos do software.

Em resumo, os modelos tradicionais oferecem estrutura e previsibilidade, enquanto os modelos ágeis proporcionam adaptabilidade e rapidez na resposta às mudanças.

REFERÊNCIAS

[Desenvolvimento de Software: Metodologias Ágeis vs. Tradicionais](#)

[Modelos Tradicionais X Métodos Ágeis - Engenharia de Software](#)

[Metodologia tradicional e metodologia ágil, qual a diferença? - Objective](#)