

Resumo expandido do modelo incremental

Slide 1 - Capa

Universidade Estadual de Minas Gerais – UEMG

Modelo Incremental

Ana Laura Silva Resende
Hérík Fernandes dos Santos
Isabela Vitória Barroso Branco
Marcos Vinícius da Silva
Thiago Henrique Campanha Almeida

Slide 2 - Sumário

- Introdução ao Modelo Incremental
- Funcionamento do Processo e Vantagens
- Objetivo Geral e Específicos
- Referencial Teórico
- Desenvolvimento e Análises
- Conclusão

Slide 3 - O que é o modelo incremental?

O modelo incremental é uma abordagem de processo de desenvolvimento de software que combina elementos do modelo em cascata com a filosofia iterativa da prototipagem. Diferente do modelo em cascata, que busca entregar o software completo em uma única fase final, o modelo incremental foca na entrega de partes funcionais do software, conhecidas como "incrementos". Cada incremento é, essencialmente, uma versão operacional e testada do produto, embora com um conjunto limitado de funcionalidades, permitindo entregas de valor mais cedo e de forma contínua ao cliente.

Slide 4 - Como funciona o processo?

O desenvolvimento é dividido em ciclos. O primeiro incremento geralmente implementa os requisitos mais essenciais e críticos do sistema. A cada novo ciclo, um novo incremento é desenvolvido e integrado à versão anterior, adicionando mais funcionalidades. Esse processo se repete até que todos os requisitos planejados para o produto sejam implementados e entregues. Cada ciclo passa por todas as fases clássicas do desenvolvimento: análise de requisitos, projeto, codificação e teste, garantindo a qualidade de cada parte entregue.

Slide 5 - Objetivo geral do modelo

O objetivo geral do modelo incremental é mitigar os riscos associados a projetos de longa duração, permitindo a entrega rápida e progressiva de um software funcional. A meta central é fornecer valor ao cliente nos estágios iniciais do ciclo de vida do projeto e obter feedback constante para garantir um alinhamento maior entre o produto final e as necessidades reais do usuário.

Slide 6 - Objetivos específicos

Para alcançar seu objetivo principal, o modelo incremental busca atender aos seguintes objetivos específicos:

- Reduzir o tempo para a entrega da primeira versão funcional do software.
- Facilitar a coleta de feedback do cliente em estágios iniciais e contínuos.

- Distribuir e gerenciar os riscos do projeto ao longo de seu ciclo de vida.
- Aumentar a flexibilidade para acomodar mudanças nos requisitos do sistema.
- Simplificar o gerenciamento da complexidade ao decompor o projeto em partes menores.

Slide 7 - Como funciona o modelo incremental

Abordagem de desenvolvimento de software baseada em entregas sucessivas.

Slide 8 - Conceito

O sistema é desenvolvido em partes funcionais chamadas incrementos. Cada incremento adiciona novas funcionalidades ao software.

Slide 9 – Ideia principal

O produto final é entregue aos poucos. O cliente já pode usar as primeiras versões. O sistema vai evoluindo gradualmente.

Slide 10 – Funcionamento

1. Identificação dos requisitos principais
2. Criação do primeiro incremento (versão simples)
3. Entrega ao cliente para uso e feedback
4. Inclusão de novos incrementos até completar o sistema

Slide 11 – Etapas

Planejamento inicial Análise e projeto do incremento Implementação e testes Entrega e integração ao sistema

Slide 12 – Estrutura visual

Incremento 1: funções básicas Incremento 2: novas funcionalidades Incremento 3: expansão do sistema Ao final, o software está completo

Slide 13 – Exemplo

1ª entrega: módulo de login e cadastro 2ª entrega: módulo de relatórios 3ª entrega: módulo de pagamento O sistema cresce passo a passo até a versão final

Slide 14 - Vantagens do modelo incremental

Esse modelo é especialmente útil quando não há disponibilidade de equipe suficiente para realizar uma implementação completa dentro do prazo estipulado pelo projeto. A divisão em partes facilita o cumprimento dos prazos sem sobrecarregar os recursos disponíveis.

O cliente não precisa esperar o sistema inteiro estar pronto para começar a utilizá-lo. Como as funcionalidades são entregues conforme a prioridade, as mais importantes ficam disponíveis primeiro, permitindo que o cliente já comece a tirar proveito do sistema.

Slide 15 - Vantagens do modelo incremental

A necessidade de retrabalho em termos de análise e documentação é consideravelmente menor em comparação com o modelo em cascata, o que reduz o tempo gasto com revisões e atualizações.

O modelo incremental facilita a coleta de feedback dos clientes, já que eles podem acompanhar o progresso do desenvolvimento por meio de versões executáveis do sistema, em vez de

apenas documentos técnicos, o que normalmente é difícil de avaliar para quem não é da área técnica.

Slide 16 - Vantagens do modelo incremental

É possível entregar rapidamente uma versão funcional do software, mesmo que ainda não tenha todas as funcionalidades previstas. Isso permite que o cliente comece a usar o sistema e obter benefícios logo nas primeiras fases do projeto.

O modelo oferece maior flexibilidade e capacidade de adaptação. Mudanças nos requisitos podem ser incorporadas de maneira mais tranquila nos próximos incrementos, sem comprometer o desenvolvimento já realizado.

Slide 17 - Desvantagens do modelo incremental

A aplicação do modelo pode ser problemática em sistemas de grande porte, especialmente quando são complexos, têm longa vida útil ou são desenvolvidos por várias equipes ao mesmo tempo.

Em projetos grandes, é essencial que exista uma arquitetura sólida desde o início, além de uma clara definição das responsabilidades de cada equipe. Como aponta **sommerville2011engenharia**, essa base arquitetural não pode ser construída de forma incremental e deve ser planejada com antecedência.

Slide 18 - Desvantagens do modelo incremental

Um dos desafios do modelo é a dificuldade em acompanhar o progresso do projeto. Gerentes e stakeholders costumam depender de entregas visíveis e regulares para avaliar o andamento do desenvolvimento. Como os incrementos podem ser rápidos, nem sempre é viável documentar cada versão do sistema de forma detalhada.

Outro problema é a degradação da estrutura do sistema ao longo do tempo. À medida que novos incrementos são adicionados, sem uma visão clara do todo, a arquitetura pode se tornar instável. Isso exige tempo e recursos para refatorar e manter a integridade do software, pois a incorporação constante de mudanças pode dificultar sua evolução e aumentar os custos de manutenção.

Slide 19 - Desenvolvimento incremental

Características:

- Melhoria do desenvolvimento cascata. Tudo é organizado e especificado no início, bem documentado de ponta a ponta;
- Primeiramente tem o levantamento de requisitos. Mas o desenvolvimento baseado na entrega de “incrementos” partes;
- Atividades são intercaladas;
- Objetivo: dar feedback rápido ao cliente;
- Entrega incrementos. Vai fazendo por partes e vai entregando ao cliente;
- Fácil de acomodar mudança, no próximo incremento você apenas altera;
- Começa pela parte mais fácil;
- O processo pode não ser muito claro;
- O sistema não é completamente especificado no início.

Slide 20 - Análise SWOT: forças (strenghts)

O desenvolvimento incremental apresenta como principal força a flexibilidade. Ele permite que requisitos sejam ajustados ao longo do processo, reduzindo o risco de retrabalho em larga escala e aproximando o produto final da real necessidade do usuário. Além disso, a entrega rápida de versões parciais gera valor imediato, fortalece a confiança do cliente e facilita a coleta de feedback contínuo. Outra força é a capacidade de priorizar funcionalidades críticas logo nos primeiros ciclos, possibilitando que a empresa entregue resultados tangíveis antes de concluir o sistema completo.

Slide 21 - Análise SWOT: fraquezas (weaknesses)

As fraquezas do modelo se manifestam principalmente em projetos de grande porte ou de vida longa. A arquitetura tende a se degradar com o acúmulo de incrementos, exigindo disciplina em refatoração, o que pode aumentar custos ao longo do tempo. Outro ponto frágil é a

menor visibilidade para gestores que dependem de documentação formal para acompanhar o progresso, já que o processo incremental foca mais em entregas rápidas do que em relatórios detalhados.

Slide 22 - Análise SWOT: oportunidades (opportunities)

As oportunidades para o incremental estão alinhadas às tendências modernas de software, como metodologias ágeis, integração contínua e DevOps. Empresas de tecnologia que precisam acompanhar mudanças rápidas no mercado digital encontram nesse modelo um suporte adequado para lançar produtos inovadores, validar hipóteses e escalar serviços em nuvem. Também há oportunidade no uso de feedback analítico de usuários, que pode ser integrado a cada ciclo para direcionar a evolução do software.

Slide 23 - Análise SWOT: ameaças (threats)

As ameaças, por sua vez, aparecem quando há dependência de arquitetura robusta. Em sistemas críticos, como aviação ou saúde, mudanças incrementais mal controladas podem comprometer a confiabilidade e a segurança. Além disso, em cenários corporativos muito regulados, a falta de documentação formal exigida por auditorias ou órgãos normativos pode inviabilizar a adoção plena do incremental. Há ainda o risco de atrasos se a priorização das funcionalidades não for bem definida, levando a versões iniciais pouco úteis para os clientes.

Slide 24 - Análise SWOT

Slide 25 - Análise comparativa entre modelos: incremental, cascata e RAD

Quando se compara o modelo incremental com o modelo em cascata e o RAD(Rapid Application Development) , a primeira diferença aparece na relação com os requisitos. O cascata exige que todos sejam definidos de forma completa antes do início do desenvolvimento, funcionando melhor em sistemas estáveis, como softwares de folha de pagamento governamentais ou

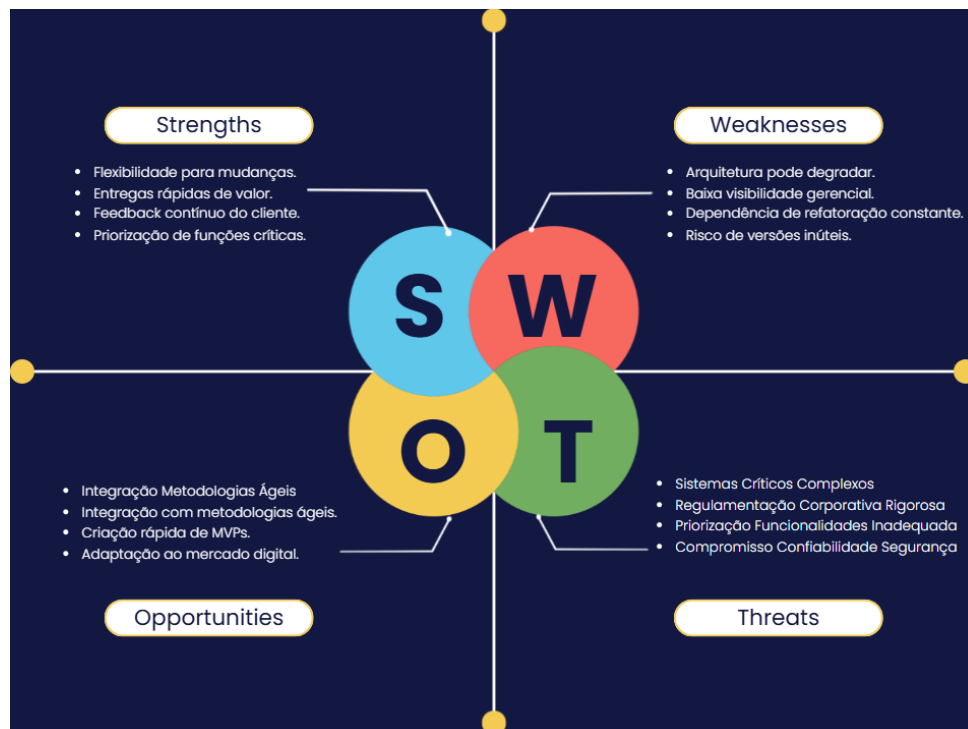


Figura 1: Representação visual da análise SWOT

sistemas embarcados em automóveis que devem ser planejados de ponta a ponta. Já o incremental admite ajustes ao longo do desenvolvimento, sendo mais adequado a contextos dinâmicos, como aplicativos de e-commerce. O RAD, por sua vez, valoriza a prototipação rápida, dependendo de requisitos iniciais apenas como guia, e evolui com base em interações intensas com o usuário, sendo útil em projetos que precisam de resultados visíveis em curto prazo, como sistemas de suporte interno para empresas.

Slide 26 - Análise comparativa entre modelos: incremental, cascata e RAD

A segunda diferença está no tempo de entrega. O cascata disponibiliza o produto somente ao final de todas as fases, o que pode levar meses ou anos. O incremental entrega versões parciais desde cedo, acelerando o retorno de valor. O RAD vai além, buscando entregas ainda mais velozes por meio de protótipos e iterações curtas, aproximando-se de uma lógica quase experimental, onde o software evolui rapidamente até se ajustar ao esperado pelo cliente.

Outro aspecto é a visibilidade do processo. O cascata traz clareza documental, permitindo que gestores acompanhem cada fase, mas com pouca flexibilidade. O incremental oferece menos documentação formal, exigindo acompanhamento pelas entregas reais em cada ciclo. O RAD, em contrapartida, dá visibilidade por meio de protótipos e interfaces tangíveis, queaju-

dam os clientes a compreender o andamento do projeto, ainda que isso sacrifique a profundidade técnica da documentação.

Slide 27 - Análise comparativa entre modelos: incremental, cascata e RAD

Quanto ao custo de mudanças, o cascata é o mais rígido, já que alterações significam refazer fases inteiras. O incremental absorve mudanças de forma mais econômica, distribuindo ajustes ao longo dos ciclos. O RAD é ainda mais permissivo, pois as mudanças são esperadas desde o início, mas esse ritmo acelerado pode comprometer a qualidade se não houver disciplina técnica.

Na arquitetura de software, o cascata tem vantagem por definir uma estrutura sólida desde o começo, garantindo consistência. O incremental corre risco de degradação arquitetural se não houver refatorações planejadas. Já o RAD pode negligenciar a arquitetura em nome da velocidade, criando sistemas pouco escaláveis ou difíceis de manter em longo prazo.

Slide 28 - Análise comparativa entre modelos: incremental, cascata e RAD

Em relação ao cliente, o cascata o envolve basicamente no início e no fim. O incremental promove participação frequente, a cada ciclo. O RAD exige envolvimento constante e intenso, com usuários colaborando diretamente na criação e validação dos protótipos, o que pode ser vantajoso em pequenos grupos, mas inviável em projetos de grande porte com muitos stakeholders.

Por fim, analisando riscos e adaptação ao mercado, o cascata concentra riscos no final, quando erros são mais caros. O incremental dilui riscos ao longo dos ciclos, corrigindo falhas gradualmente. O RAD reduz riscos de desalinhamento com o cliente, pois ele participa ativamente, mas pode gerar riscos técnicos relacionados à pressa no desenvolvimento. Em termos de mercado, o cascata se encaixa em setores regulados e de baixa mudança, o incremental em negócios digitais em evolução constante, e o RAD em projetos que precisam de soluções rápidas e prototipação intensa.

Slide 29 - Tabela comparativa: modelos incremental, cascata e RAD

MODELOS DE DESENVOLVIMENTO: ENGENHARIA DE SOFTWARE			
TABELA COMPARATIVA			
CARACTERÍSTICAS	INCREMENTAL	CASCATA	RAD
REQUISITOS PODEM MUDAR	✓	✗	✓
ADEQUADO PARA REQUISITOS ESTÁVEIS	✗	✓	✗
ENTREGAS PARCIAIS	✓	✗	✓
PRODUTO SÓ NO FINAL	✗	✓	✗
RETORNO RÁPIDO DO INVESTIMENTO	✓	✗	✓
FORTE DOCUMENTAÇÃO	✗	✓	✗
FOCO EM PROTOTIPAGEM RÁPIDA	✗	✗	✓
CLIENTE PARTICIPA EM CICLOS CONTÍNUOS	✓	✗	✓
ARQUITETURA DEFINIDA DESDE O INÍCIO	✗	✓	✗
ADEQUADO PARA MERCADO DINÂMICO	✓	✗	✓

Figura 2: Tabela comparativa entre as características dos 3 modelos

Slide 30 - Conclusão sobre o modelo incremental de engenharia de software

O modelo incremental se consolidou como uma abordagem robusta e flexível no ciclo de vida de desenvolvimento de software, representando uma evolução significativa em relação a modelos mais rígidos e sequenciais. Sua principal característica reside na entrega do software em partes funcionais, ou incrementos, permitindo que o cliente receba valor de forma antecipada e contínua ao longo do projeto. Cada incremento passa por todas as fases da engenharia de software, desde o levantamento de requisitos até os testes, garantindo que ao final de cada ciclo uma nova porção funcional do sistema esteja operacional.

Slide 31 - Conclusão sobre o modelo incremental de engenharia de software

Os principais pontos que definem o modelo incremental incluem a sua natureza iterativa, a entrega progressiva de funcionalidades e a capacidade de adaptação a mudanças. Essas características se traduzem em vantagens competitivas importantes. A principal delas é a mitigação de riscos, uma vez que os problemas podem ser identificados e corrigidos em estágios iniciais do desenvolvimento, ao invés de serem descobertos apenas na fase final de testes. Além disso, a flexibilidade para acomodar novos requisitos ou alterar os existentes é uma vantagem marcante, pois o planejamento detalhado é realizado para o incremento atual, permitindo ajustes nos ciclos subsequentes. A entrega antecipada de um produto funcional, mesmo que com um escopo reduzido, possibilita a obtenção de feedback valioso do usuário final, alinhando o produto final às suas reais necessidades.

Slide 32 - Conclusão sobre o modelo incremental de engenharia de software

Contudo, o modelo incremental não está isento de desvantagens. Uma crítica comum é a possibilidade de degradação da arquitetura do sistema se não houver um esforço consciente de refatoração a cada novo incremento. A adição contínua de funcionalidades pode levar a um design de sistema pouco coeso e de difícil manutenção a longo prazo. Outro desafio reside na necessidade de um planejamento e integração cuidadosos, para garantir que os diferentes incrementos se conectem de forma harmoniosa, evitando inconsistências.

Slide 33 - Breve comparação com o modelo em cascata

Em contraposição ao modelo incremental, o modelo em cascata adota uma abordagem estritamente sequencial e linear. Nele, cada fase do desenvolvimento (requisitos, projeto, implementação, testes e implantação) deve ser completamente concluída antes que a próxima possa ser iniciada. Essa rigidez torna o modelo em cascata inadequado para projetos onde os requisitos são voláteis ou não são bem compreendidos no início. A principal desvantagem do modelo em cascata é a sua inflexibilidade e o fato de que um erro detectado em fases avançadas pode ter um custo de correção altíssimo, exigindo um retorno a fases muito anteriores do projeto. Enquanto o modelo incremental preza pela entrega de valor contínuo e pela adaptação, o modelo

em cascata foca na conclusão de fases bem definidas, entregando o produto completo apenas ao final do processo.

Em suma, a escolha pelo modelo incremental é particularmente acertada para projetos que se beneficiam da entrega rápida de funcionalidades, onde os requisitos podem evoluir e o feedback do cliente é um componente crucial para o sucesso do produto final.

Slide 34 - Referências