Ciclo de Vida Do Software

Prof. Esp.
Rita de Cássia da Costa Silva

FAMA
CENTRO UNIVERSITÁRIO

Conceitos Básicos

Definição: Conjunto de fases e atividades necessárias para desenvolver, entregar e manter um software desde sua concepção até o seu fim de vida útil.

IMPORTÂNCIA

Organiza o processo de desenvolvimento.

Define entregas parciais (artefatos) em cada fase.

Reduz riscos e custos através de planejamento estruturado.

Serve como guia de comunicação entre times técnicos e de negócio.

ETAPAS DO CICLO DE VIDA

1- Planejamento
2- Análise de Requisitos
3- Design do Sistema/Projeto
4- Implementação (Codificação)
5- Testes
6- Implantação
7- Manutenção

PLANEJAMENTO

Levantamento Inicial De Necessidades.

Definição De Objetivos, Escopo E Recursos.

PRODUTO/ARTEFATO

Documento de visão do projeto, cronograma inicial.

Documento de Visão

I. Comunica a ideia central do produto de software.

II. Define o problema que ele resolve e o público-alvo.

III. Aponta objetivos, escopo, restrições e metas iniciais.

IV. Serve para alinhar todos os stakeholders (clientes, desenvolvedores, gestores, equipe de testes, etc.).

V. Funciona como referência durante o projeto, principalmente para não desviar do objetivo inicial.

FAMA CENTRO UNIVERSITÁRIO

ESTRUTURA DO DOCUMENTO

1- Introdução

2- Descrição do Problema

3- Descrição do Produto

4- Escopo

5- Público-Alvo e Usuários

6- Objetivos e Benefícios

7- Restrições

8- Cronograma Estimado

9- Critérios de Sucesso

FAMA
CENTRO UNIVERSITÁRIO

FAMA
CENTRO UNIVERSITÁRIO

ESTRUTURA DO DOCUMENTO

1- Introdução

2- Descrição do Problema

3- Descrição do Produto

4- Escopo

5- Público-Alvo e Usuários

6- Objetivos e Benefícios

7- Restrições

8- Cronograma Estimado

9- Critérios de Sucesso

1 Contexto do Projeto
Objetivo Geral
Público-Alvo

2 Qual necessidade ou problema o software vai resolver.
Cenário atual e suas limitações

3 Ideia central da solução.
Principais funcionalidades.
Diferenciais do Produto

4 Ideia central da solução.
Principais funcionalidades.
Diferenciais do Produto

5 Faixa Etária, Nível Técnico, ...

6 Metas a serem alcançadas
Benefícios esperados

7 Orçamentárias, Tecnológicas
Prazos, ...

8 Datas Estimadas

9 Produto atingiu seus objetivos ?

FAMA CENTRO UNIVERSITÁRIO

FAMA CENTRO UNIVERSITÁRIO

ANÁLISE DE REQUISITOS

Coleta e documentação de requisitos funcionais e não funcionais.

Ferramentas: Entrevistas, Questionários, Workshops

PRODUTO/ARTEFATO

Documento de requisitos, diagramas UML

DOCUMENTO DE REQUISITOS

É uma representação visual que mostra os requisitos do sistema (funcionais e não funcionais) e as relações entre eles e outros elementos do projeto, como atores (usuários, sistemas externos), casos de uso, componentes de software ou objetivos de negócio.

DIAGRAMAS UML

É uma linguagem visual padrão para representar, especificar e documentar sistemas de software.

Diagrama de Caso de Uso

- Mostra as interações entre *atores* (usuários/sistemas externos) e o sistema.
- Funcionalidades esperadas

Diagrama de Classes

- Mostra a estrutura do sistema: classes, atributos, métodos e relações.
- Organização de Dados e Responsabilidades.

Diagrama de Sequência

- Mostra a troca de mensagens entre objetos/atores ao longo do tempo.
- Comportamento Dinâmico e Ordem De Eventos.

FAMA CENTRO UNIVERSITÁRIO

FAMA CENTRO UNIVERSITÁRIO

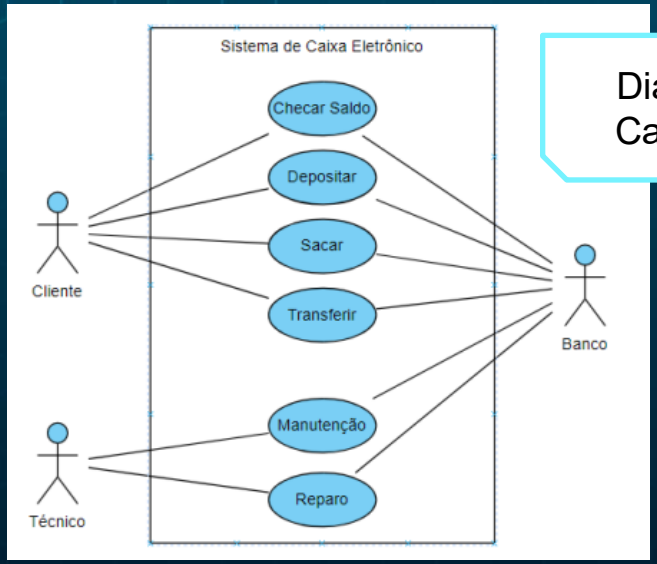


Diagrama de Caso de Uso

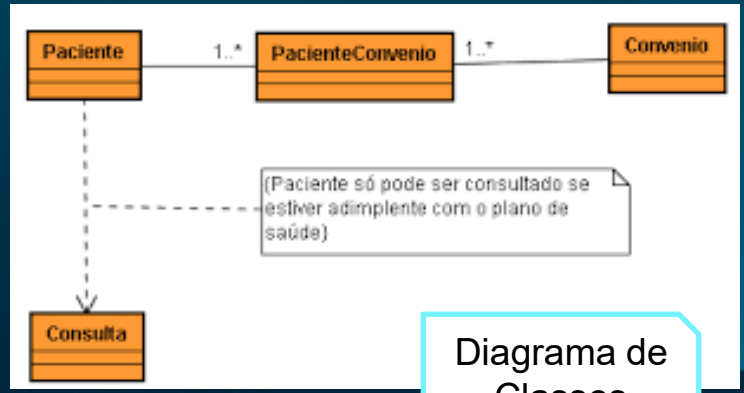


Diagrama de Classes

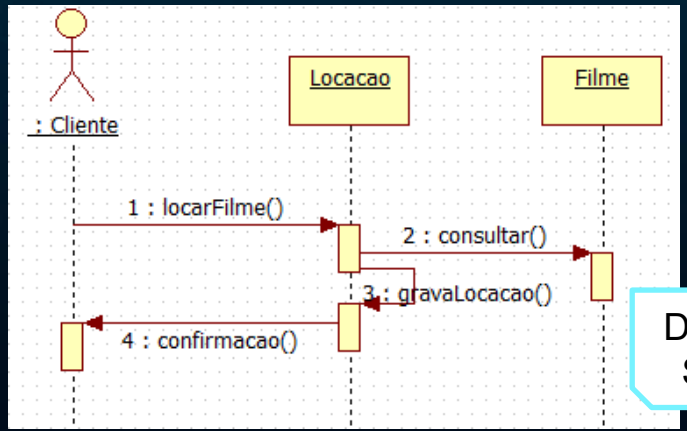


Diagrama de Sequência

FAMA CENTRO UNIVERSITÁRIO

FAMA CENTRO UNIVERSITÁRIO

DESIGN DO SISTEMA/PROJETO

Arquitetura do sistema, escolha de tecnologias, protótipos.

PRODUTO/ARTEFATO

Diagramas de arquitetura, protótipos de telas.

DIAGRAMA DE ARQUITETURA

É uma representação visual de como os componentes de um sistema estão organizados e como se comunicam. Mostra módulos, serviços, bancos de dados, interfaces externas e fluxos de dados.

Arquitetura em Camadas

- Mostra a separação de responsabilidades. Bom para sistemas monolíticos ou modulares.

Arquitetura de Microsserviços

- Mostra cada serviço como um bloco independente. Bom para sistemas escaláveis e distribuídos.

Arquitetura de Nuvem

- Mostra recursos em provedores. Bom para sistemas com um grande volume de dados.

Arquitetura de Componentes

Detalha como cada componente interno interage. Bom para sistemas embarcados ou desktop.

FAMA CENTRO UNIVERSITÁRIO

IMPLEMENTAÇÃO (CODIFICAÇÃO)

Tradução do projeto em código. Uso de boas práticas da programação.

Ferramentas: Entrevistas, Questionários, Workshops

PRODUTO/ARTEFATO

Código-fonte, Commits no Repositório.

TESTES

Verificação da qualidade, detecção de bugs.

Tipos: unitário, integração, sistema, aceitação.

PRODUTO/ARTEFATO

Relatórios de teste.

IMPLANTAÇÃO & MANUTENÇÃO

Disponibilização do software ao usuário

PRODUTO/ARTEFATO

Versões publicadas, Documentação de Manutenção.

Ferramentas Diagramas

Draw.io

Lucidchart

creately

Excalidraw

Modelos Tradicionais

Prof. Esp.
Rita de Cássia da Costa Silva

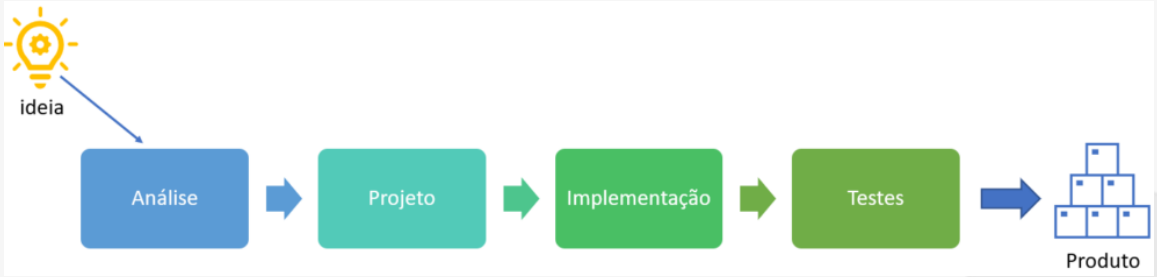
FAMA
CENTRO UNIVERSITÁRIO

Modelos utilizados para desenvolver o ciclo de vida do software.

- É linear e rígido, com pouca possibilidade de voltar atrás sem retrabalho.
- Pouca adaptação a mudanças no meio do projeto.
- Risco de descobrir problemas tarde demais (só nos testes).
- Feedback do cliente chega depois de meses, quando já é caro corrigir.

SITUAÇÕES IDEIAIS DE USO:

- Útil quando o escopo e requisitos são claros e estáveis desde o início.
- Favorece projetos que precisam de muita documentação e aprovação formal antes de cada fase.
- Quando é possível definir 100% das funcionalidades antes de codificar.

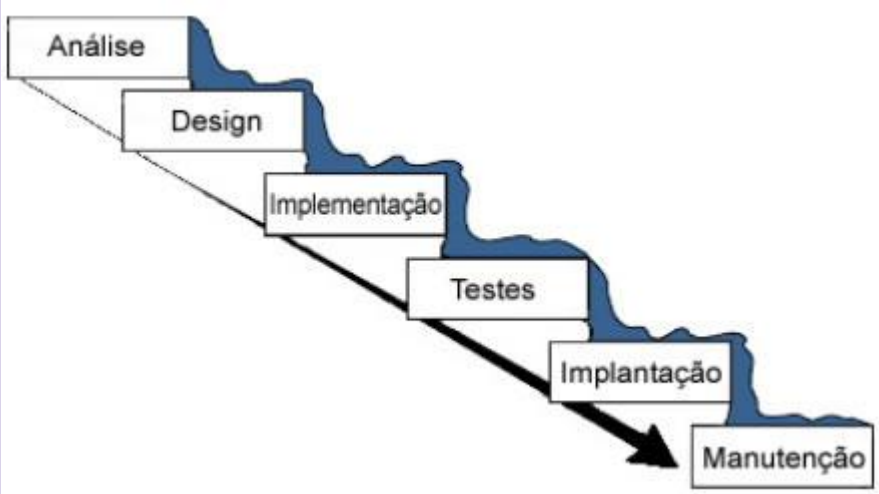


MODELO **CASCATA**

- Funciona em Fases Sequenciais: Só se avança para a próxima etapa após concluir a anterior.

COMO TRABALHAR COM ESTE MODELO:

- Coletar todos os requisitos antes de iniciar o projeto.
- Criar documentação completa para guiar toda a implementação.
- Realizar testes somente ao final.



Exemplo
Software para controle de inventário de um museu

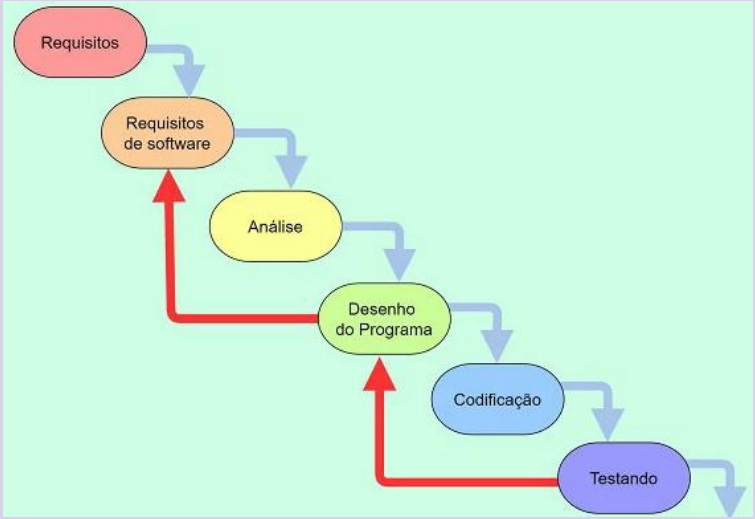
Requisitos claros, não há mudanças frequentes, foco na estabilidade e documentação para auditoria.

MODELO **CASCATA COM RETORNO**

- Similar ao cascata, mas permite voltar uma ou duas fases quando há erros ou mudanças pequenas (mais flexível).

COMO TRABALHAR COM ESTE MODELO:

- Adotar revisões periódicas ao final de cada fase.
- Permitir pequenos ajustes sem reiniciar todo o processo.



Exemplo
Sistema acadêmico para universidade

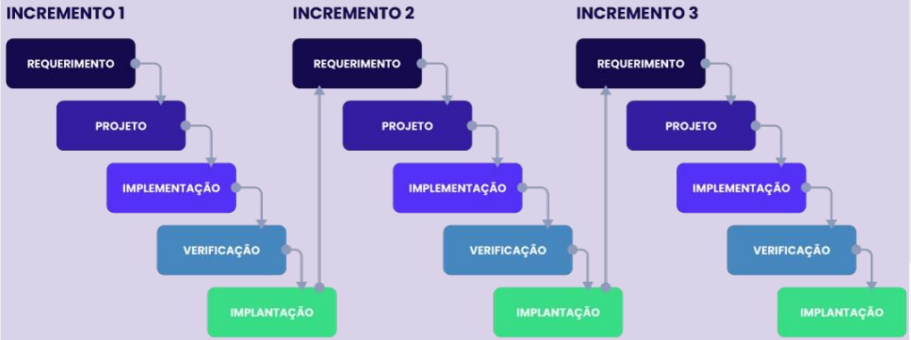
Requisitos estáveis, mas possibilidade de ajustes pontuais após validação do protótipo.

MODELO INCREMENTAL

- Entrega do sistema em módulos: Cada incremento passa pelas fases do ciclo de vida, mas adiciona funcionalidades gradualmente.

COMO TRABALHAR COM ESTE MODELO:

- Dividir o projeto em funcionalidades prioritárias
- Entregar um núcleo básico e adicionar recursos em incrementos posteriores.



Exemplo
Aplicativo de delivery

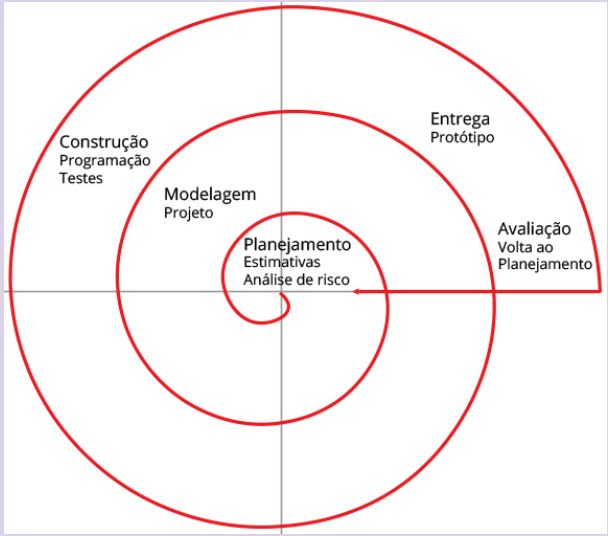
Primeira versão com cadastro e pedidos, depois chat com o entregador, rastreamento em tempo real, promoções, etc...

MODELO **ESPIRAL**

- Passa várias vezes pelo ciclo de vida, refinando o produto a cada volta.

COMO TRABALHAR COM ESTE MODELO:

- Fazer um protótipo inicial
- Revisar riscos e melhorias a cada ciclo antes de prosseguir.
- Ideal para projetos com alto risco ou requisitos incertos.



Exemplo
Sistema de controle de voo

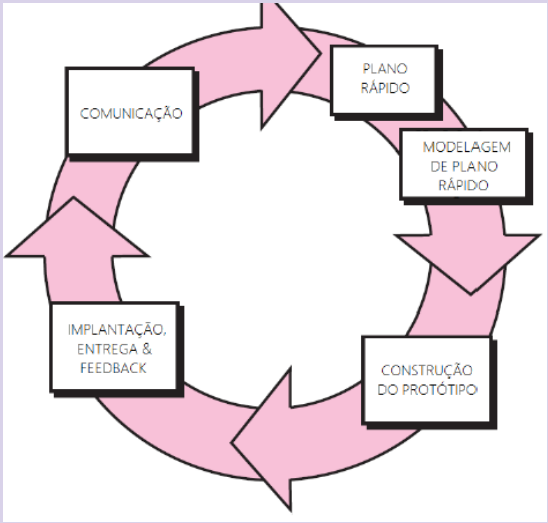
Protótipos para simular segurança e precisão, revisados com engenheiros antes da implementação final.

MODELO DE PROTOTIPAGEM

- Criação de protótipos rápidos para validar requisitos com o cliente antes do desenvolvimento completo (descartável ou evolutivo).

COMO TRABALHAR COM ESTE MODELO:

- Criar protótipos de baixa fidelidade para aprovar com usuários.
- Refinar o protótipo até fechar os requisitos, depois implementar.



Exemplo
App de rede social

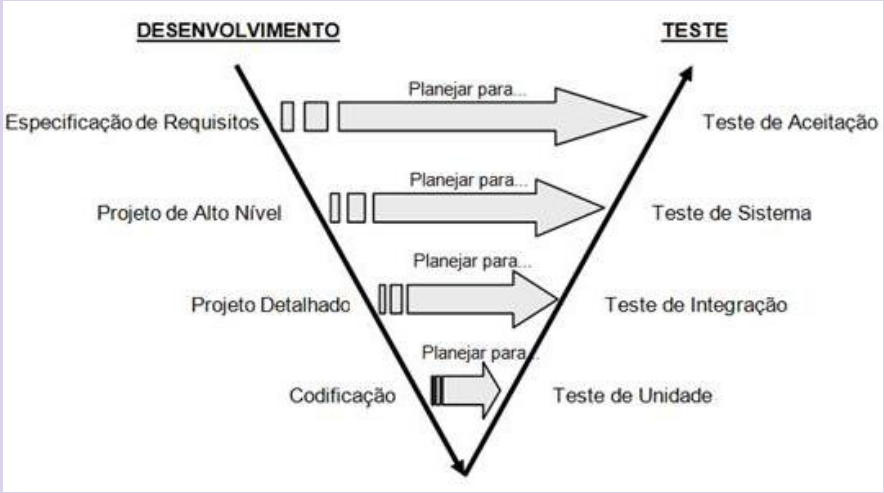
Protótipo com navegação e telas principais para validar a experiência do usuário antes da programação.

MODELO V

- Variante do cascata com testes associados a cada fase de desenvolvimento.
- O lado esquerdo é o desenvolvimento; o lado direito é o teste correspondente.

COMO TRABALHAR COM ESTE MODELO:

- Definir casos de teste já na fase de requisitos e projeto.
- Validar cada fase com o teste planejado antes.



Exemplo
Sistema médico hospitalar

Testes definidos desde o início para garantir segurança e conformidade com normas.

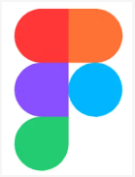
RESUMO EM TABELA

MODELO	CARACTERÍSTICA	VANTAGENS	LIMITAÇÕES
Cascata	Linear e sequencial	Clareza, documentação	Pouca flexibilidade
Cascata com Retorno	Linear com ajustes	Menos retrabalho	Ainda pouco flexível
Incremental	Entregas por partes	Entrega rápida de valor	Integração complexa
Espiral	Iterativo + riscos	Alta segurança	Mais caro
Prototipagem	Validação rápida	Reduz erros de requisitos	Pode gerar expectativa irreal
V-Model	Testes paralelos ao desenvolvimento	Qualidade alta	Alto custo inicial

Ferramentas Modelos

Draw.io

Lucidchart



Figma

Excalidraw

"Ser desenvolvedor é uma viagem onde a próxima parada é a solução de um problema."

Thales Valentim

E-mail: **rita.silva@faculdadefama.edu.br**

GitHub: **https://github.com/ritasilva-fama**

Whatsapp