

MODELOS DE PROCESSO PRESCRITIVO

Cascata
Incremental
Evolucionário
Concorrente



MODELOS DE PROCESSO



O QUE É

Define o fluxo de todas as atividades, ações e tarefas e o grau de iteração



QUEM REALIZA

Os engenheiros de software



POR QUE É IMPORTANTE

O processo propicia estabilidade, controle e organização



QUAIS AS ETAPAS ENVOLVIDAS

Modelos de processo fornecem os "passos" necessários para realizar um trabalho de engenharia de software disciplinado



QUAL É O ARTEFATO

Uma descrição personalizada das atividades e tarefas definidas pelo processo



COMO GARANTIR QUE O TRABALHO FOI FEITO CORRETAMENTE

Qualidade, cumprimento de prazos e a viabilidade a longo prazo do produto



MODELOS DE PROCESSO

COMO SURTIU

Para trazer ordem ao caos que existia na área de desenvolvimento de software

OBJETIVO

Encontrar equilíbrio entre pôr ordem ao modelo de software e ser adaptável quando as coisas mudam constantemente

MODELOS PRESCRITIVOS

OBJETIVO

Estruturar e ordenar o desenvolvimento de software

As atividades acontecem sequencialmente

DÚVIDA

Os modelos prescritivos se adequam a mudanças no software?

É possível substituir os modelos tradicionais por algo menos estruturado?

PRESCRITIVOS

A ordem e a consistência são pontos predominantes

Prescreve um fluxo de processo (fluxo de trabalho)

Acomodam as atividades genéricas apresentadas anteriormente

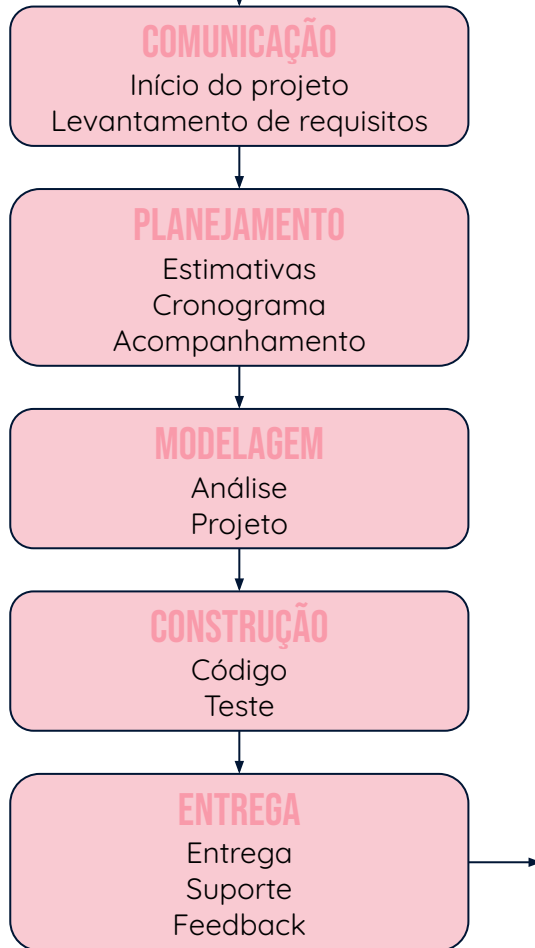


01.

MODELO

CASCATA





MODELO CASCATA

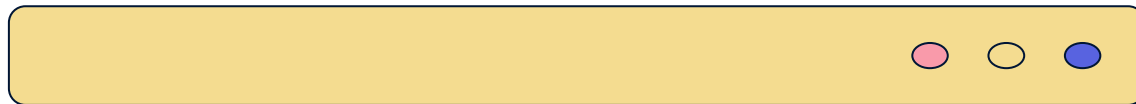
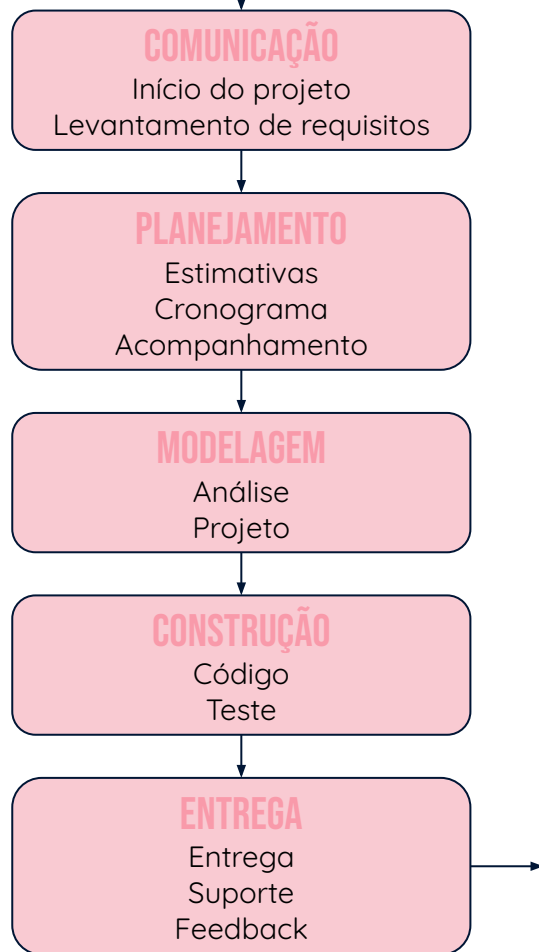
Há casos em que os requisitos do sistema são bem compreendidos

Requisitos estáveis

Menores modificações em sistemas existentes

Ciclo de vida clássico

Abordagem sequencial

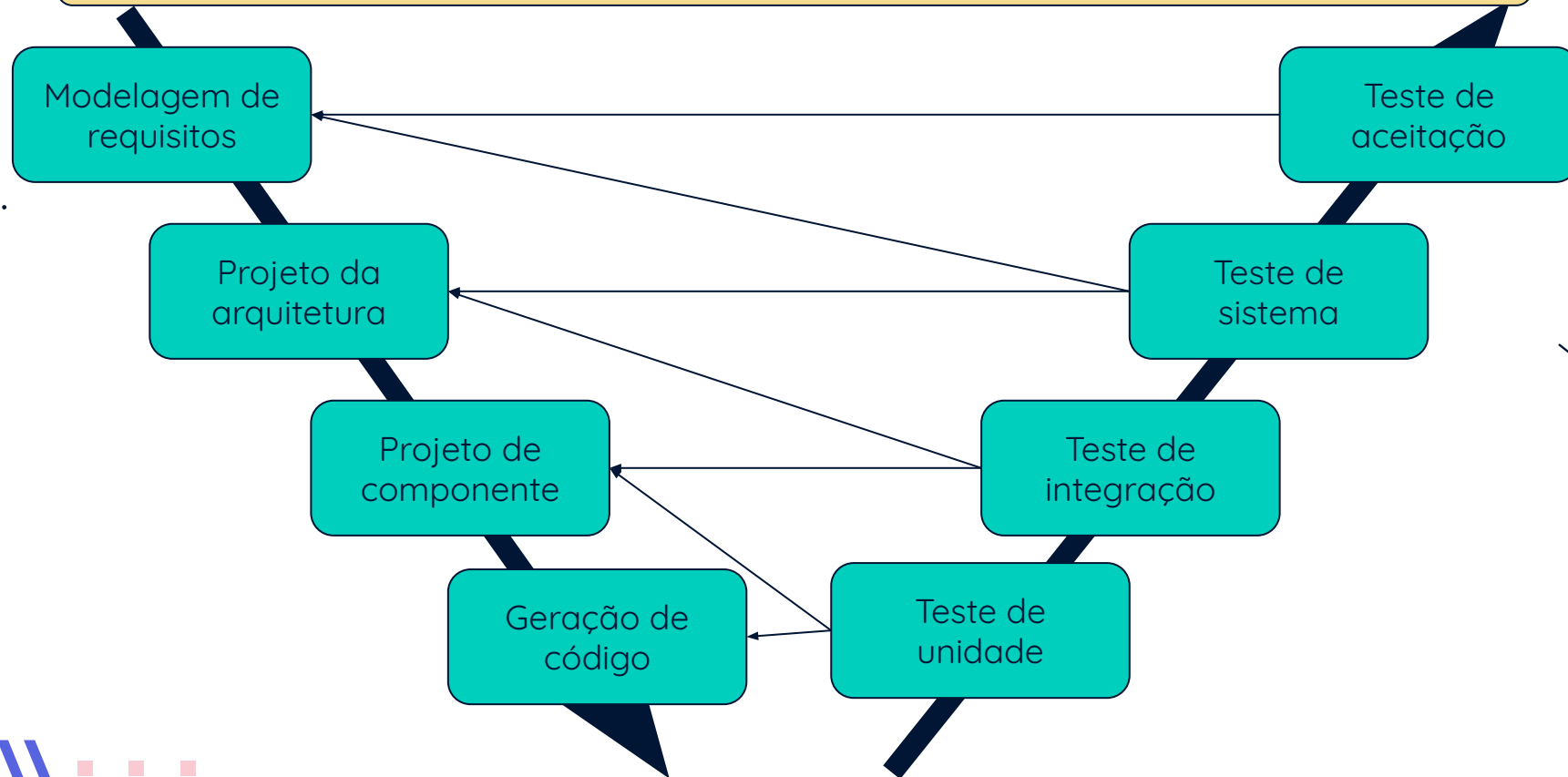


MODELO CASCATA

POSSÍVEL PROBLEMA?

1. Problemas reais nem sempre seguem esse fluxo
Pode conter iterações, mas isso pode causar confusões
2. É difícil para o cliente dizer, previamente, todas as suas necessidades
3. O cliente deve ter paciência para obter uma versão operacional do sistema
4. **Estado de bloqueio:** membros da equipe tem que aguardar outros membros da equipe
Tende a prevalecer no início e no fim do processo
5. Sistemas que possuem requisitos voláteis

MODELO CASCATA



02.

MODELO

INCREMENTAL



PRIMEIRO INCREMENTO

Produto essencial

Atende os requisitos básicos

Não entrega os recursos complementares (alguns nem foram discutidos ainda)

PRODUTO ESSENCIAL

Utilizado pelo cliente

Passar por avaliação detalhada

AValiação DO INCREMENTO

Analisa-se o próximo incremento

Processo repetido até o sistema estar completo

MODELO INCREMENTAL

Requisitos não bem definidos - impossível utilizar o fluxo linear

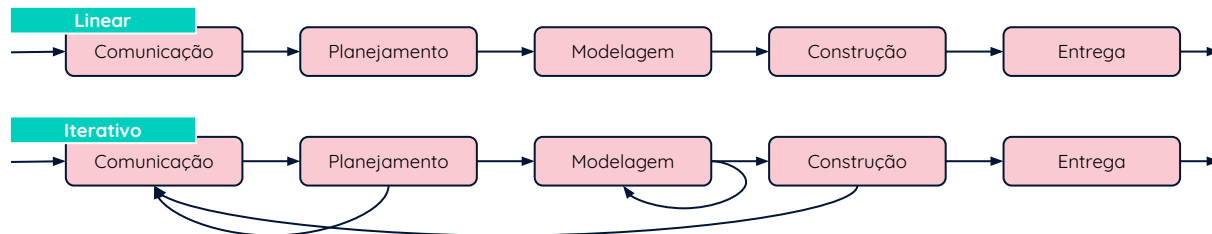
Fornecer elemento funcional aos usuários

Ajuda a definir os requisitos de forma correta

Deve-se refinar e expandir sua funcionalidade depois

É uma combinação dos fluxos linear e iterativo

Libera uma série de versões (**incrementos**) que oferecem mais funcionalidade a cada incremento



SOFTWARE DE PROCESSAMENTO DE TEXTOS

PRIMEIRO INCREMENTO

Funções básicas de gerenciamento de arquivos, edição e produção de documentos

SEGUNDO INCREMENTO

Recursos mais avançados de edição e produção de documentos

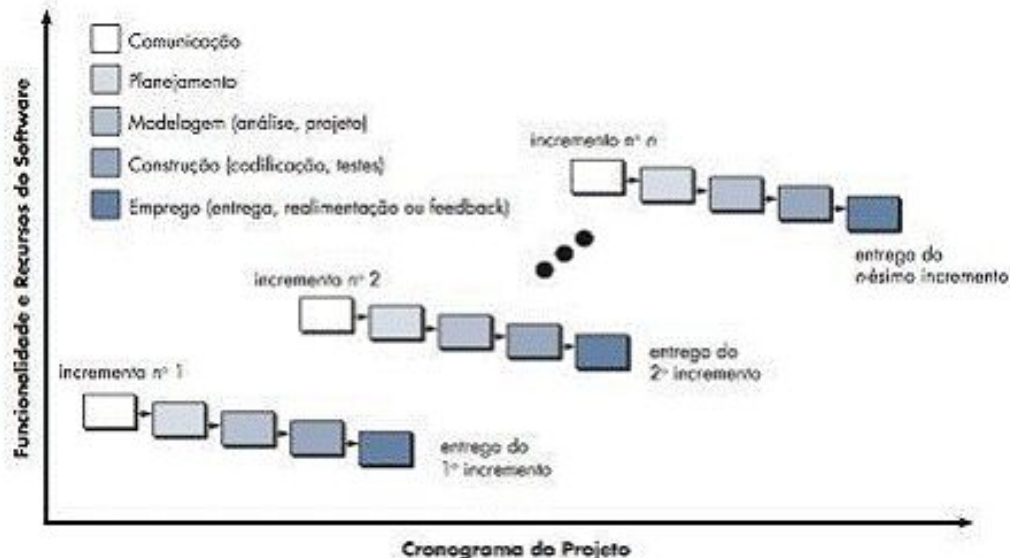
TERCEIRO INCREMENTO

Revisão ortográfica e gramatical

QUARTO INCREMENTO

Recursos avançados de formatação de página (layout)

MODELO INCREMENTAL



03.

MODELO

EVOLUCIONÁRIO



MODELOS EVOLUCIONÁRIOS



Sistemas complexos **evoluem** com o tempo

Os **requisitos tendem a mudar** conforme o projeto avança

Modelo linear passa a ser **impraticável**

Prazos apertados impostos pelo mercado geram **atrasos**

Liberar versões limitadas

Atende às pressões comerciais

Não podemos perder o timing de entrega do software

Permite analisar melhor o sistema

Os clientes aceitam bem esse modelo?

O produto é construído aos poucos, sempre com uma versão funcional em uso.

Foca na evolução gradual conforme novas necessidades surgem, em vez de tentar definir tudo de uma vez.

Usam 2 modelos comuns: prototipação e modelo espiral



PROTOTIPAÇÃO

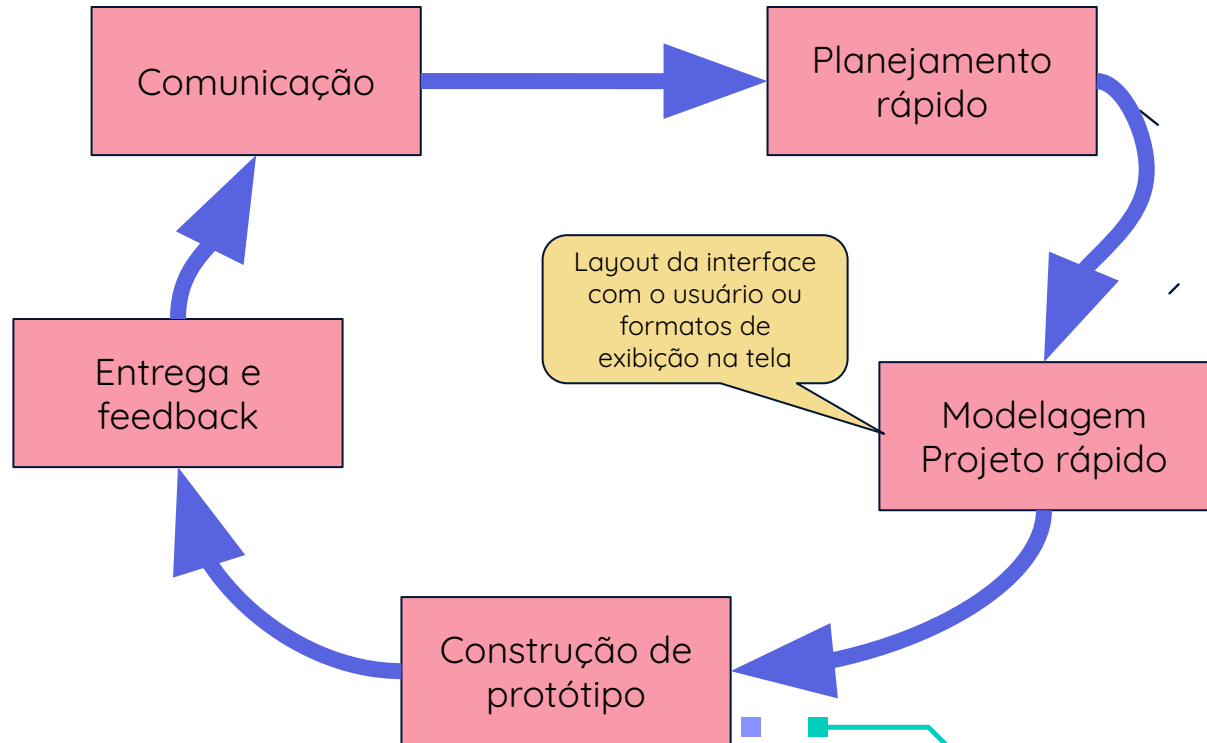
QUANDO USAR

Cliente define objetivos gerais do software

Desenvolvedor está inseguro com relação a algum algoritmo, ou à IHC, ou ao uso de um determinado SO

Requisitos estão obscuros

Pode ser um processo só, ou dentro de um determinado processo



PROTOTIPAÇÃO

TIPOS

Protótipo operacional ou não (vertical ou horizontal)

O QUE FAZER COM ELE DEPOIS

Depende da qualidade

Pode ou não ser reutilizado (como base para o projeto final)

PROBLEMAS

Clientes querem que o protótipo já seja utilizado

Caso não tenha qualidade, pode ser um problema a longo prazo

MODELO ESPIRAL

Proposto por Barry Boehm (1988)

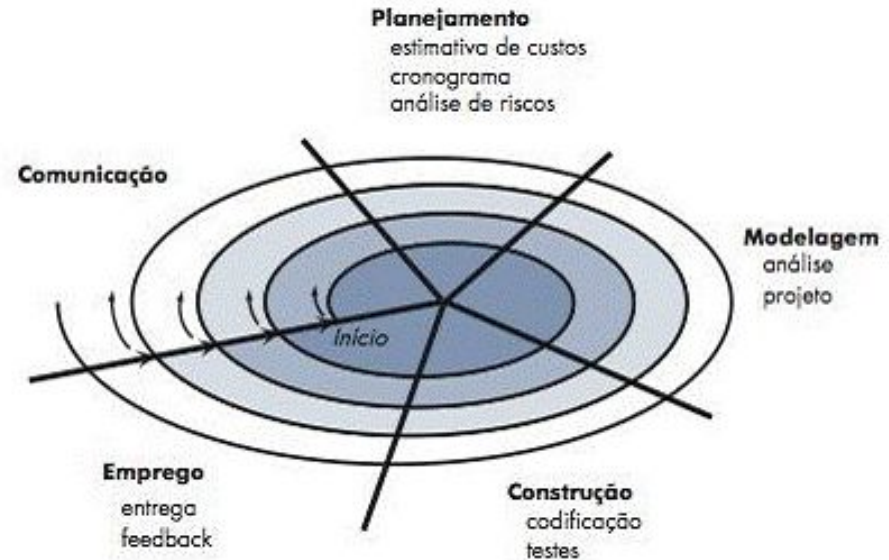
Une prototipação com modelo em cascata

Software desenvolvido em várias versões

Primeiras versões: modelo ou protótipo

Feedback permite atualizar custo e cronograma

No **planejamento** se analisa a quantidade de iterações



MODELO ESPIRAL

Não termina ao fim do projeto

Podem haver espirais para projeto, protótipo, desenvolvimento de versões, manutenção, etc.

Abordagem mais realista

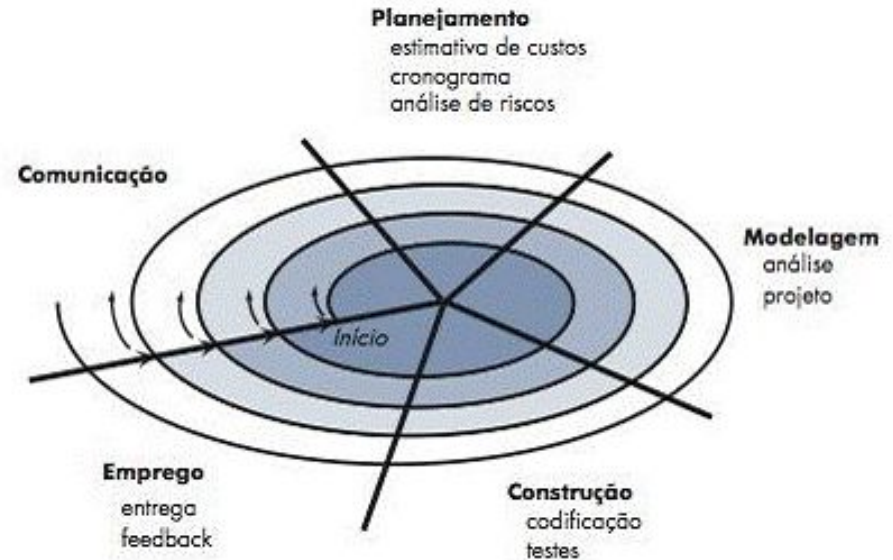
Usa prototipação em qualquer etapa do projeto

Difícil de controlar?

Analisa riscos da cada rodada

Passa por todas as etapas do PDS

É mais caro e demorado, pois tem um planejamento extenso



Aspecto	Prototipação	Modelo Espiral
Objetivo principal	Validar requisitos e ideias rapidamente criando versões simplificadas do sistema.	Reduzir riscos e planejar de forma iterativa, incorporando análise de riscos em cada ciclo.
Forma de trabalho	Criar um protótipo → mostrar ao usuário → coletar feedback → ajustar → repetir até aprovação.	Passar por ciclos (espirais) de planejamento → análise de riscos → desenvolvimento/teste → avaliação do cliente.
Foco	Comunicação com o usuário e refinamento dos requisitos.	Gestão de riscos e desenvolvimento controlado.
Uso de protótipos	Essencial, pode ser descartável ou evolutivo.	Usado como parte do ciclo, mas não obrigatório em todas as fases.
Flexibilidade	Alta – mudanças são esperadas e incorporadas facilmente.	Alta, mas sempre dentro de um planejamento e análise de riscos.
Pontos fortes	Feedback rápido, evita erros de entendimento.	Minimiza riscos, adequado para projetos grandes e complexos.
Pontos fracos	Pode gerar código bagunçado se o protótipo for aproveitado sem reestruturação.	Mais caro e demorado, exige experiência em análise de riscos.
Exemplo prático	Criar um rascunho interativo de um app para o cliente testar antes de desenvolver de fato.	Criar um sistema de controle de tráfego aéreo, planejando e validando cada parte com foco na segurança e riscos.

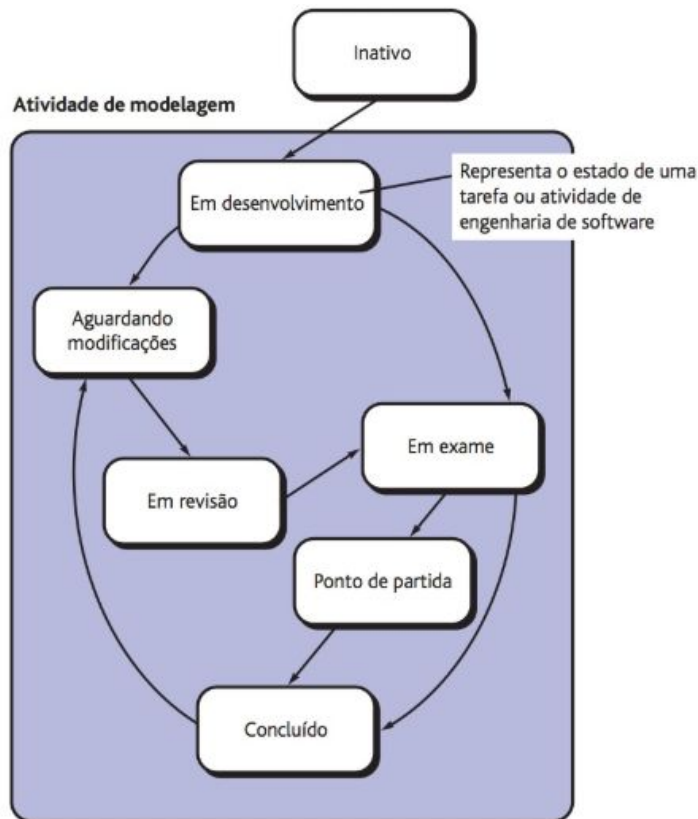
04.

MODELO

CONCORRENTE



MODELO CONCORRENTE



Representa os elementos concorrentes e iterativos de qualquer um dos modelos descritos anteriormente

As atividades de engenharia de software existem simultaneamente, cada uma em um estado

INÍCIO DO PROJETO

Comunicação → aguardando modificações

Modelagem → nenhum → em desenvolvimento

Cliente quer fazer modificações nos requisitos

Modelagem → em desenvolvimento → aguardando modificações

COMO FUNCIONA

Eventos realizam transições entre os estados

PROCESSO EVOLUCIONÁRIO

CARACTERÍSTICAS DOS SOFTWARES MODERNOS

Contínuas modificações, prazos apertados, necessidade de satisfação do cliente-usuário

Não podemos perder o tempo de entrega do produto (tempo de mercado)

DESVANTAGENS

A prototipação pode trazer dificuldades para o planejamento (quantos ciclos serão necessários?)

Com que velocidade as integrações devem ser entregues?

Tempo de acomodação X Velocidade

Deve-se garantir a qualidade!

VAMOS REFLETIR...

MODELO EM CASCATA

Vamos descrever 3 projetos que poderiam ser feitos com o modelo em cascata

MODELO DE PROTOTIPAÇÃO

Vamos descrever 3 projetos que poderiam ser feitos com o modelo de prototipação

MODELO INCREMENTAL

Vamos descrever 3 projetos que poderiam ser feitos com o modelo incremental



VAMOS REFLETIR...



PROTÓTIPO

Que adaptações
deveríamos fazer
em um protótipo
para ele virar um
sistema ou produto
a ser entregue?

SOFTWARE "BOM O SUFICIENTE"

Como definir o
ponto entre software
entregue
rapidamente e com
qualidade?



PARA DISCUSSÃO

1. Se o seu cliente só pode falar com a equipe no início e no final do projeto, qual modelo você escolheria? Por quê?
2. Entre Cascata e Incremental, qual se adapta melhor a mudanças de requisitos? Dê um exemplo real.
3. O modelo Espiral é mais caro e demorado que o Cascata. Em que tipo de projeto esse custo extra se justifica?
4. Em um projeto de sistema bancário crítico, qual modelo prescritivo usaria? E em um aplicativo de entretenimento?
5. Quais riscos você vê em usar um modelo Concorrente sem experiência prévia na equipe?

PARA DISCUSSÃO

6. Você já participou (ou conhece) um projeto que falhou por má escolha do modelo de desenvolvimento? Qual foi o problema?
7. É possível combinar elementos de dois modelos prescritivos em um mesmo projeto? Dê um exemplo de como isso funcionaria.
8. Se os requisitos mudam constantemente, existe algum modelo prescritivo capaz de lidar bem com isso? Por quê?
9. Como você explicaria para um cliente leigo a diferença entre Cascata e Incremental sem usar termos técnicos?
10. Se tivesse que escolher um modelo para desenvolver um sistema de controle de tráfego aéreo, qual seria? Por quê?

DÚVIDAS?

Qualquer dúvida pode entrar em contato:

Profª. Carla Fernandes Curvelo
Email: carla.fernandes@ufrn.br

