MODELOS DE PROCESSO

ESPECIALIZADO E UNIFICADO °







MODELO DE PROCESSO ESPECIALIZADO

Incluem características dos modelos prescritivos



DESENVOLVIMENTO BASEADO EM

COMPONENTES



01.

O QUE TEMOS NESSA ESTRATÉGIA?



REUTILIZAÇÃO DE SOFTWARE

03.

REDUÇÃO DO CUSTO DO PROJETO

02.

REDUÇÃO DO TEMPO DE DESENVOLVIMENTO

04.

AUMENTO DA Produtividade

COMO ISSO FUNCIONA?



REUTILIZAÇÃO DE ARTEFATOS JÁ CONSTRUÍDOS

02.

CONTRATAÇÃO DE PESSOAL EXPERIENTE

03.

CONSTRUÇÃO DE SISTEMAS ATRAVÉS DE COMPONENTES

COMPONENTES









COTS

Componentes de software comercial de prateleira

Possuem funcionalidades que podem facilmente ser adicionadas a outro software



MODELO

Incorpora características do modelo espiral

É evolucionário



COMO UTILIZÁ-LOS

Módulos de softwares convencionais

Classes orientadas a objeto

Pacotes de classes

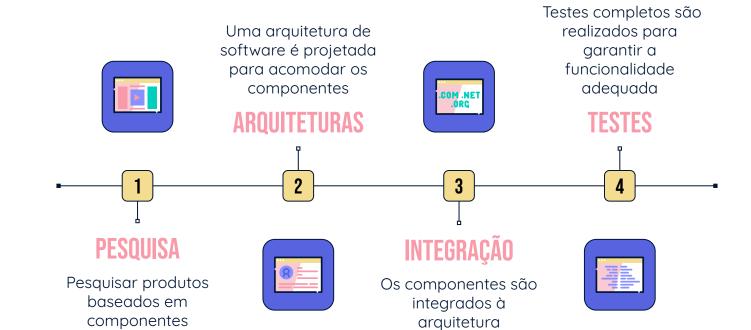
disponíveis

ETAPAS









 \circ

02.

MODELOS DE VEIUNA FIIKM712



MODELO DE MÉTODOS FORMAIS



- Atividades que conduzem uma especificação matemática formal do software
- Uma das variações da abordagem: engenharia de software sala limpa
- Remover ambiguidade, incompletude e inconsistência através de uma análise matemática
 - Não utiliza revisões locais
- Não é uma das abordagens mais utilizadas
 - a. Consome muito tempo e dinheiro
 - b. Precisa de uma formação específíca
 - Difícil de explicar os métodos formais aos clientes

EXEMPLOS







- Realizar provas matemáticas que garantem que este modelo possui as propriedades requisitadas (verificação).;
- Analisar se a solução proposta é aceitável do ponto de vista de desempenho, indicando quais as melhores estratégias para implementação a serem seguidas;
- Validar o modelo através de simulações;
- Realizar o desenvolvimento do software podendo-se provar que a implementação está correta (geração de código correto).

Criar uma modelagem

formal

Usa teoria de conjuntos ou

álgebra de processos

0

MODELO DE MÉTODOS FORMAIS



foram atendidos

Prova de teorema

automatizada ou verificação

de modelo



através dos

componentes gráficos



ORIENIAUU A ASPECTOS



DESENVOLVIMENTO DE SOFTWARE ORIENTADO A ASPECTOS • •

- Questões como segurança, log, transações, etc são tratadas através de aspectos
- O processo de desenvolvimento deles pode seguir um modelo concorrente ou evolucionário

Decomposição

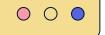
Separação dos requisitos funcionais e transversais (infraestrutura, serviços)

Implementação

Implementação dos requisitos em módulos separados

Recomposição

Aspectos → Módulos que implementam as regras de recomposição do sistema Processo de "costura" **04**.



PROCESSO

UNIFICADO

Idealizado pelos criadores da UML



"Hoje, o software tende em direção a sistemas maiores e mais complexos. Isso se deve, em parte, ao fato de que os computadores se tornam mais potentes a cada ano, aumentando a expectativa dos usuários em relação a eles. Essa tendência também tem sido influenciada pelo uso crescente da Internet para troca de todos os tipos de informação... Nosso apetite por software cada vez mais sofisticado aumenta à medida que tomamos conhecimento de como um produto pode ser aperfeiçoado de uma versão para a seguinte. Queremos software que seja cada vez mais adequado às nossas necessidades, mas isso, por sua vez, torna o software cada vez mais complexo. Em suma, queremos cada vez mais."

-IVAR JACOBSON, GRADY BOOCH E JAMES RUMBAUGH

PROCESSO UNIFICADO





É uma tentativa de aproveitar os melhores recursos e características dos modelos tradicionais de processos de software, mas caracterizando-os de modo a implementar muitos dos melhores princípios de desenvolvimento ágil de software.

Importância de comunicação com o cliente

Métodos para descrever a visão do cliente (casos de uso)

> Perspectiva do usuário (língua natural)

Dividir o problema

Ajuda o arquiteto a manter metas como compreensão, confiança em mudanças futuras e reutilização

Incremental e iterativo

PROCESSO UNIFICADO

Dirigido a casos de uso
Centrado na arquitetura
Iterativo
Incremental



PROCESSO UNIFICADO









Início dos anos 90

James Rumbaugh, Grady Booch, Ivar Jacobson queriam criar um "método unificado"

Criaram a UML (linguagem de modelagem unificada)

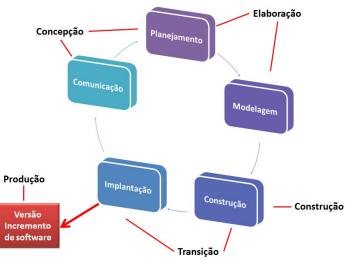
Por volta de 1997 a UML se tornou um padrão da indústria

CONCEPÇÃO









Inclui a comunicação com o cliente e o planejamento

Etapa curta

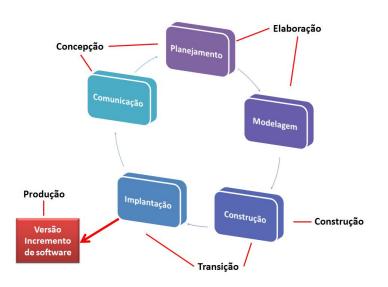
- l. Identificar as necessidades de negócio
- Propor uma arquitetura rudimentar para o sistema
- 3. Planejar a natureza iterativa e incremental do projeto
- 4. Criar um diagrama de casos de uso

CONCEPÇÃO









Arquitetura

Um esquema provisório dos principais subsistemas, funções e recursos

Deve ser refinada posteriormente

Planejamento

Identifica recursos

Avalia riscos

Define cronograma

Define uma base das fases que existirão no sistema

ELABORAÇÃO

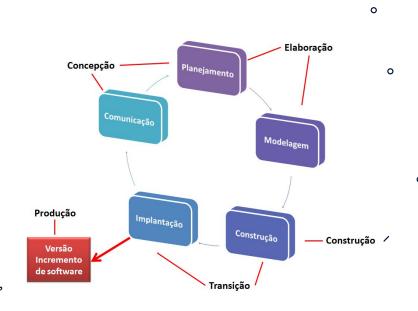






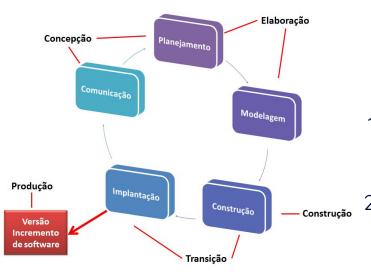
Inclui o planejamento e a modelagem do processo

- Refina e expande os casos de uso preliminares
- 2. Amplia a visão arquitetural:
 - a. Modelo de casos de uso
 - b. Modelo de análise
 - c. Modelo de projeto
 - d. Modelo de implementação
 - e. Modelo de disponibilização
- O plano é revisado, para garantir escopo, riscos e datas para entrega adequados e realistas



CONSTRUÇÃO





Inclui a atividade de construção

Entrada: modelo de arquitetura

- Implementa-se todas as funcionalidades necessárias para o incremento de software
 - Realizam-se atividades de integração (montagem de componentes e testes de integração)

TRANSIÇÃO



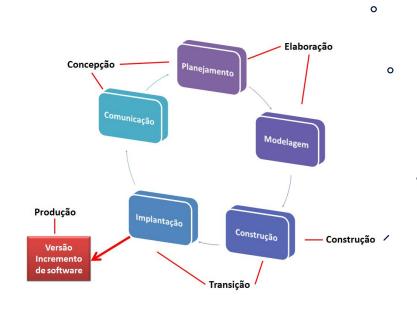




Inclui a construção e a implantação do processo

- Entrega-se o software para os usuários para testes beta
 - a. O feedback dos testes permite realizar mudanças no sistema
- Preparação de material de apoio (manuais, quias, procedimentos de instalação)

Conclusão: o incremento é uma versão utilizável do software

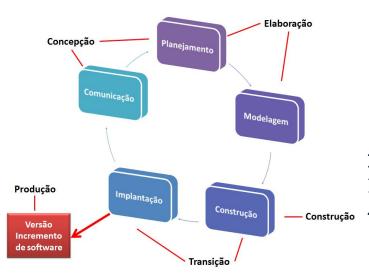


PRODUÇÃO



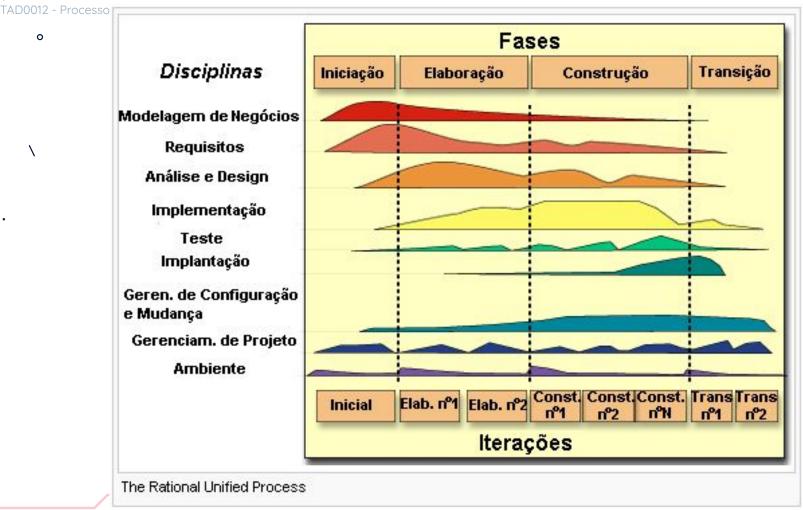






Inclui a atividade de entrega

- 1. Monitoramento do uso do sistema
- 2. Disponibilização de suporte
- 3. Realização e análise de relatórios
- 4. Análise da solicitação de mudanças



05.

IMPLEMENTAÇÃO DE

PROCESSO DE

SOFTWARE



COMO IMPLEMENTAR UM NOVO PROCESSO DE SOFTWARE? • • •

- Não é fácil mudar o processo de desenvolvimento de software de uma empresa
 - Apenas instalá-la
 - b. Assistir um curso rápido
 - c. Notificar a equipe
- Afeta como a equipe realiza o seu trabalho
- Deve-se fazer um planejamento e gerenciamento sobre as mudanças que serão realizadas

DÚVIDAS?

Qualquer dúvida pode entrar em contato:

Prof^a. Carla Fernandes Curvelo

Email: carla.fernandes@ufrn.br

