



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPI JOÃO CÂMARA

RATIONAL UNIFIED PROCESS - RUP

Nickerson Fonseca Ferreira
nickerson.ferreira@ifrn.edu.br

Introdução

2

- Modelo de processo de software proprietário.
- Desenvolvido pela empresa Rational Software Corporation.
- Em 2003 a empresa foi adquirida pela IBM. Então o RUP ganhou um novo nome: IRUP.
- Utiliza a abordagem da orientação a objetos.
- Projetado e documentado utilizando notações da UML.
- Utiliza práticas e técnicas aprovadas no mercado.

Introdução

3

- Está muito próximo da Engenharia de Software baseada em componentes.
- Possui sua preocupação baseada no atendimento do negócio.
- Possui um conceito de processo de software híbrido.
 - Traz elementos de todos os modelos de processo de software.
 - Baseado em iterações.
 - Ilustra boas práticas de especificação e projeto.

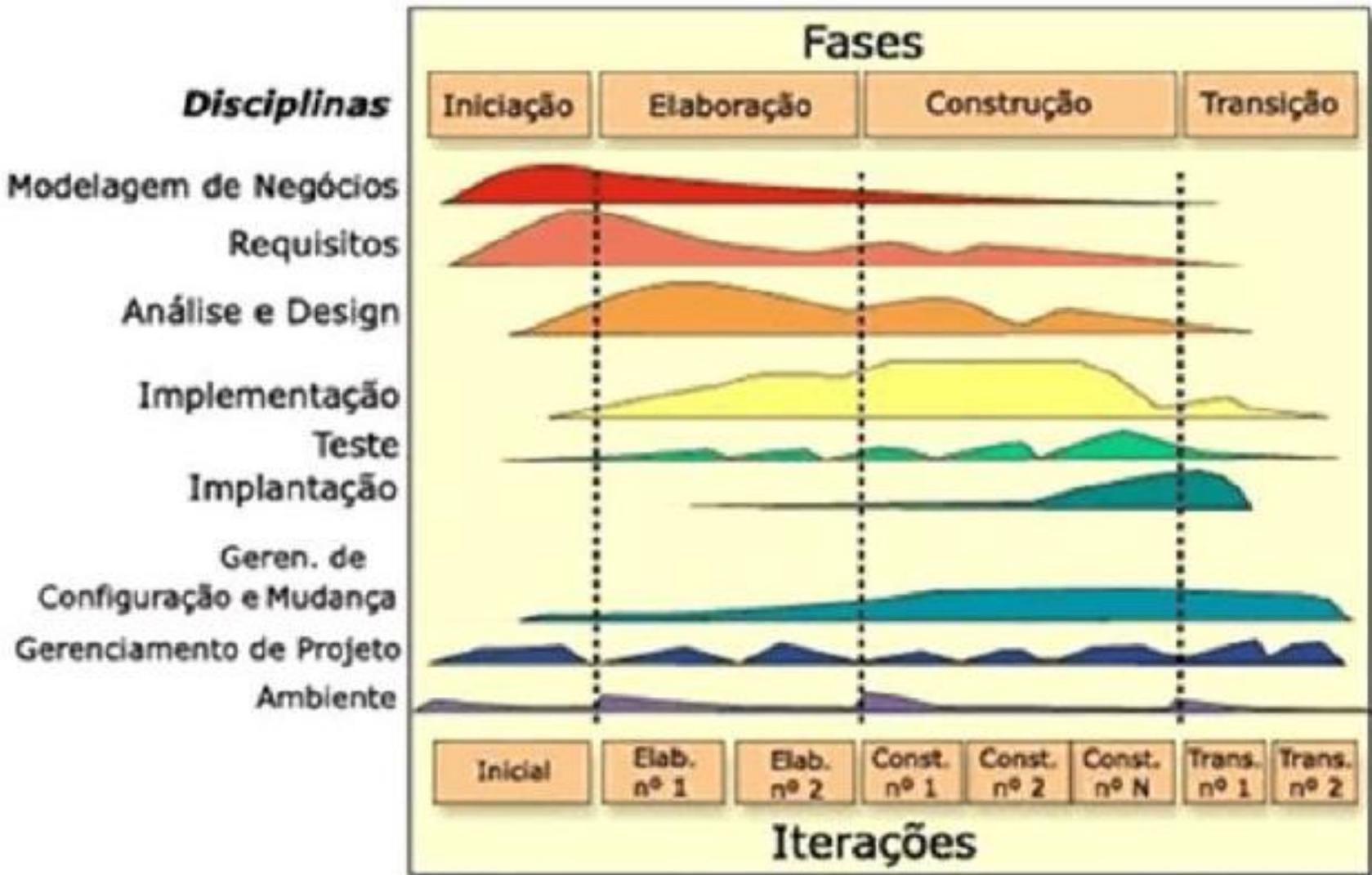
RUP

4

- Descrito a partir de 3 perspectivas:
 - **Dinâmica**: são as fases do modelo ao longo do tempo.
 - **Estática**: são as atividades realizadas no processo.
 - **Prática**: boas práticas a serem utilizadas durante o processo.

RUP

5



Fases do RUP (Dinâmica)

6

- O modelo é constituído por 4 fases:
 - Concepção
 - Elaboração
 - Construção
 - Transição

Fases do RUP (Dinâmica)

7

- **Concepção**: estabelecer um business case para o sistema.
- Avaliação do ambiente de negócio em relação à contribuição de um sistema para o negócio (**ESCOPO**).
- Identificar também as entidades externas: pessoas e sistemas, que irão interagir com o sistema.
- Depois utilizamos essas informações para avaliar a importância do sistema com o negócio.

Fases do RUP (Dinâmica)

8

- Os objetivos da fase **Elaboração** são:
 - Desenvolver um entendimento do domínio do problema;
 - Estabelecer um framework de arquitetura para o sistema;
 - Desenvolver o plano de projeto;
 - Identificar os riscos do projeto.
- Planejamento.

Fases do RUP (Dinâmica)

- **Construção:** está diretamente ligada ao projeto, programação e testes. (**DESENVOLVIMENTO**)
- Partes do sistema são implementadas paralelamente e integradas durante essa fase.
- O resultado dessa fase é um software funcional e toda sua documentação pronta para ser liberada para os usuários.

Fases do RUP (Dinâmica)

10

- **Transição**: é a última fase do RUP.
- Transferência do desenvolvimento para o usuário.
- Entrada do sistema em produção.
- Fase onerosa e problemática.
- **Implantação**

Disciplinas ou Workflows (Estática)

11

- Foca nas atividades que ocorrem durante o processo de desenvolvimento.
- Também conhecidas como WORKFLOWS.
- Todos os workflows podem ser ativados em todos os estágios do processo.
- Existem no total 9 workflows:
 - 6 workflows de processo
 - 3 workflows de apoio

Disciplinas ou Workflows (Estática)

- **Modelagem de negócio**: processo de negócio são modelados utilizando casos de uso.
- **Requisitos**: identificar os agentes que interagem com o sistema e os casos de uso são desenvolvidos.
- **Análise e projeto**: modelo de projeto é criado e documentado utilizando os modelos de arquitetura, modelos de componente, modelos de objeto e modelos de sequência.
- **Implementação**: componentes de sistema são implementados e estruturados em subsistemas de implementação.

Disciplinas ou Workflows (Estática)

- **Teste**: processo iterativo realizado em conjunto com a implementação.
- **Implantação**: versão final do produto é criada, distribuída aos usuários e instalada.
- **Gerenciamento de configuração e mudanças**: controla as mudanças do sistema.
- **Gerenciamento de projetos**: gerencia o desenvolvimento do sistema.
- **Ambiente**: relacionado à disponibilização de ferramentas apropriadas de software para a equipe de desenvolvimento.

Linhas Mestras (Prática)

14

- O RUP recomenda 6 práticas fundamentais (ou linhas mestras):
 1. Desenvolver o software de forma iterativa: incrementos de software priorizados e entregues.
 2. Gerenciar requisitos: documentação e acompanhamento das mudanças dos requisitos.
 3. Usar arquitetura baseada em componentes: estruturar a arquitetura de sistema de componentes.

Linhas Mestras (Prática)

15

4. Modelar o software graficamente: modelos gráficos da UML.
5. Verificar a qualidade do software: atender aos padrões de qualidade da organização.
6. Controlar mudanças do software: utilizando um SGM e procedimentos bem definidos.