

Ygor Amaral <ygor.amaral@ufrpe.br>

Disciplina: Processo de Desenvolvimento de Software

Curso: Sistemas de Informação (2016.1)



- A UML forneceu a tecnologia necessária para dar suporte à prática de engenharia de software orientada a objetos
  - Mas não ofereceu a metodologia de processo para orientar as equipes de projeto na aplicação da tecnologia
- Devido a isso, o RUP foi criado!
  - Uma metodologia para engenharia de software orientada a objetos usando a UML



- É uma tentativa de aproveitar os melhores recursos e características dos modelos tradicionais de processo de software
  - Mas caracterizando-os de modo a implementar muitos dos melhores princípios do desenvolvimento ágil de software
- Dirigido a planos
  - Com ideias ágeis....
  - É ideal apenas para projetos orientado a objetos
    - Pois é orientado a diagramas UML, que são orientados a objetos
    - Guiado principalmente por casos de uso
      - Requisitos



- Reconhece a importância:
  - Da comunicação com o cliente
  - De métodos racionalizados (sequencializados)

- ...para descrever a visão do cliente sobre um sistema
  - Os casos de uso!



Enfatiza o importante papel da arquitetura de software

- Ajuda o arquiteto a manter o foco nas metas corretas:
  - Compreensibilidade
  - Confiança em mudanças futuras
  - Reutilização



- É um exemplo de modelo de processo moderno, derivado de trabalhos sobre a UML
  - É um bom exemplo de processo híbrido

- Reúne elementos de outros modelos de processo
  - Apoia:
    - O desenvolvimento de documentação abrangente
    - A entrega incremental



- Sugere um fluxo de processo iterativo e incremental
  - Proporcionando a sensação evolucionária que é essencial no desenvolvimento de software moderno

#### Iterativo e Incremental



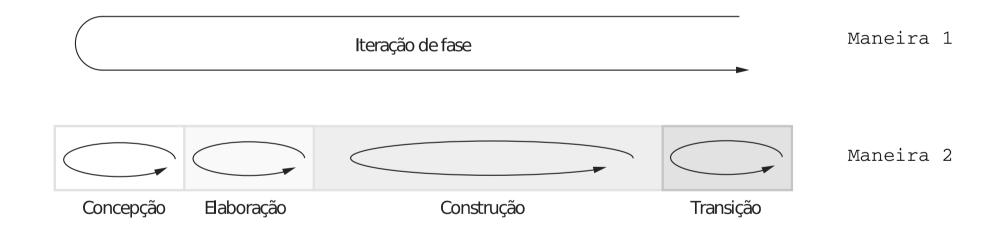
- No RUP, a iteração é apoiada de duas maneiras:
  - Cada fase pode ser executada de forma iterativa com os resultados desenvolvidos de forma incremental

 Além disso, todo o conjunto de fases também pode ser executado de forma incremental

Podem existir N iterações

## Iterativo e Incremental





## Guiado por casos de uso



 A escolha dos casos de uso é baseada em uma análise dos riscos envolvidos no projeto

- Os casos de uso que apresentam os maiores riscos devem ser realizados primeiro
  - Para resolver os riscos o quanto antes!

## Guiado por casos de uso



- Casos de uso são usados para especificar requisitos
- Durante a análise, projeto e implementação os casos de uso são "realizados"

 Durante os testes, verifica-se se o sistema realiza o que está descrito no Modelo de Casos de Uso

 Casos de uso são usados no planejamento e acompanhamento das iterações

## Guiado por casos de uso



Casos de uso são usados durante todo o processo:

Requisitos	Análise e Projeto	Implemen- tação	Testes	Implantação
Casos	de Uso fazem a ligação entre essas etapas			s etapas

# Baseado na arquitetura do sistema



- A arquitetura é desenvolvida e definida logo nas primeiras iterações
- O desenvolvimento consiste em complementar a arquitetura
  - A arquitetura guia o projeto e implementação das diversas partes do sistema
- A arquitetura serve para:
  - Organizar o desenvolvimento
  - Estruturar a solução
  - Aproveitar oportunidades de reuso

## Baseado na arquitetura do sistema



A arquitetura é o alicerce do sistema

- Quando desenvolvida com rigor:
  - Minimiza riscos técnicos
  - Favorece o reuso de componentes
  - Estimula o moral da equipe

## Orientado a objetos

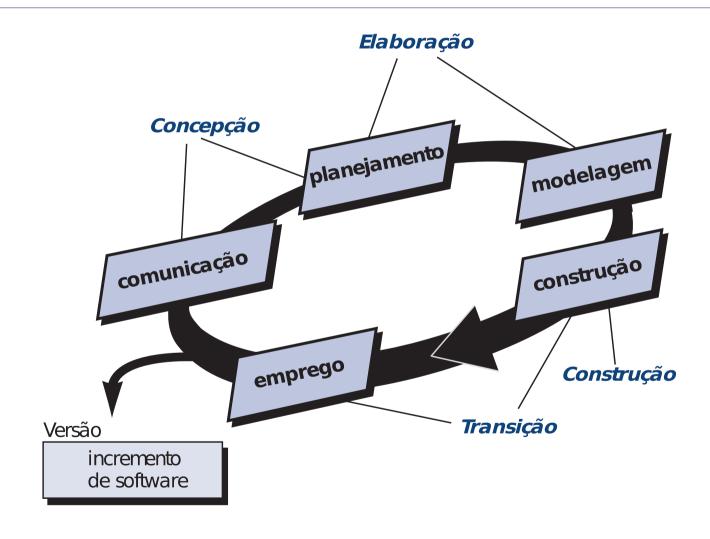


- Análise e Projeto em UML
  - UML é uma linguagem usada para especificar, modelar e documentar os artefatos de um sistema
    - Possui diversos diagramas:
      - Classes
      - Casos de uso
      - Sequência
      - Interação
  - Implementação em Java ou alguma outra linguagem de programação orientada a objetos



- É um modelo constituído de fases que identifica quatro fases distintas no processo de software:
  - Concepção
  - Elaboração
  - Construção
  - Transição







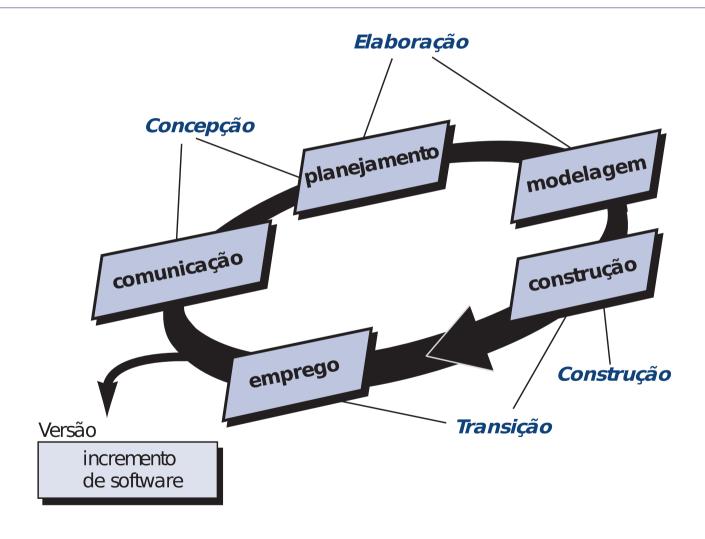
#### Concepção

- O objetivo da fase de concepção é estabelecer um business case para o sistema
- Envolve tanto a atividade de comunicação com o cliente como a de planejamento
  - Identificam-se as necessidades de negócio para o software
  - Propõe-se uma arquitetura rudimentar para o sistema
  - Desenvolve um planejamento para a natureza iterativa e incremental do projeto decorrente



- Concepção
  - Você deve identificar todas as entidades externas (pessoas e sistemas)
    - Que v\u00e3o interagir com o sistema
    - Definir casos de uso
    - Definir as interações e requisitos
  - Você deve usar essas informações para avaliar a contribuição do sistema para o negócio
    - Se essa contribuição for pequena, então o projeto poderá ser cancelado depois dessa fase
  - Em suma, estabelecer o escopo e viabilidade econômica do projeto







#### Elaboração

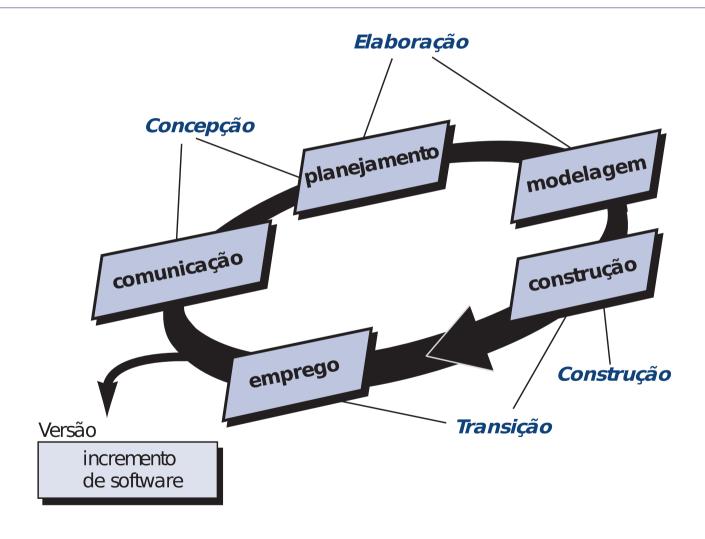
- As metas da fase de elaboração são:
  - Desenvolver uma compreensão do problema dominante
  - Desenvolver o plano do projeto
  - Identificar os maiores riscos do projeto
  - Continuar a definição de requisitos e casos de uso
  - Estabelecer a arquitetura para o sistema
    - Iniciar os testes do sistema



#### Elaboração

- No fim dessa fase, você deve ter um modelo de requisitos em estágio avançado para o sistema, que pode ser:
  - Um conjunto de casos de uso da UML
  - Plano de desenvolvimento do software
  - Uma descrição da arquitetura
  - Arquitetura principal codificada







#### Construção

- Envolve:
  - Projeto
  - Programação
  - Testes do sistema

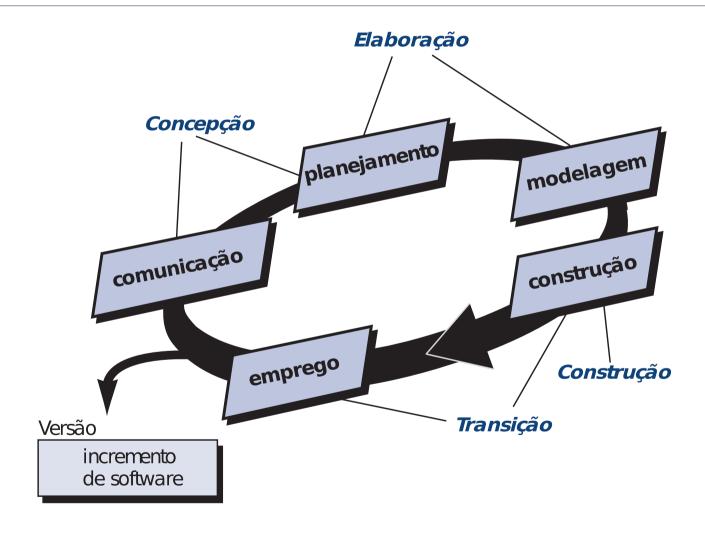
- Com a arquitetura pronta, essa fase continua o desenvolvimento do sistema
  - É a principal fase dedicada a implementação do sistema



#### Construção

- Durante essa fase, as partes do sistema são desenvolvidas em paralelo e integradas
  - Desenvolver o produto até que ele esteja pronto para testes
  - Realiza os testes...
- Na conclusão dessa fase, você deve ter um sistema de software já funcionando
  - Normalmente na versão beta
  - Também precisa ter feito a documentação associada pronta para ser entregue aos usuários







#### Transição

#### É a fase final do RUP

- Realiza os testes e correções finais...
- Implica a transferência do sistema da comunidade de desenvolvimento para a comunidade de usuários
  - Implantação
  - Nesse momento o seu funcionamento precisa acontecer em um ambiente real

#### Na conclusão dessa fase:

- Você deve ter um sistema de software documentado e funcionando corretamente em seu ambiente operacional
  - Release final
  - Não pode ser mais beta



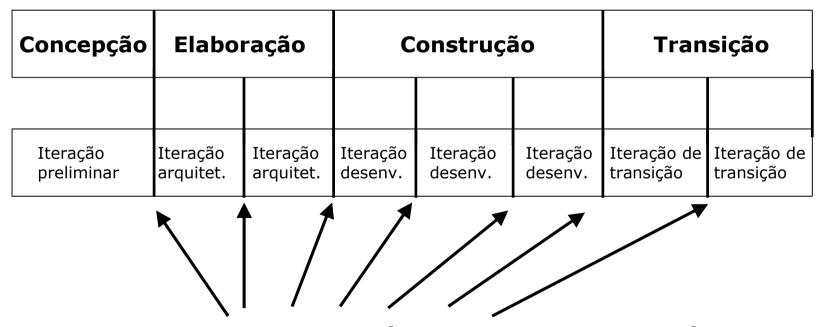
#### • Consiste de quatro fases:



26/04/2016 Rational Unified Process 28



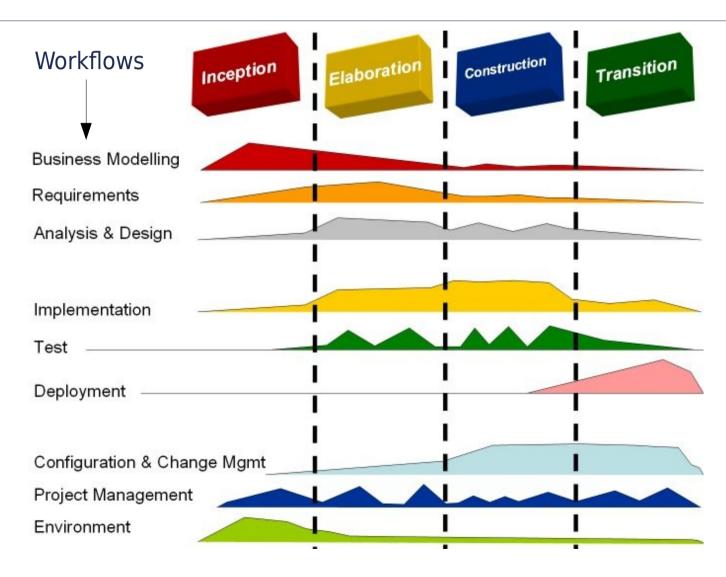
#### • Cada fase é dividida em iterações:



Marcos secundários: releases intermediários

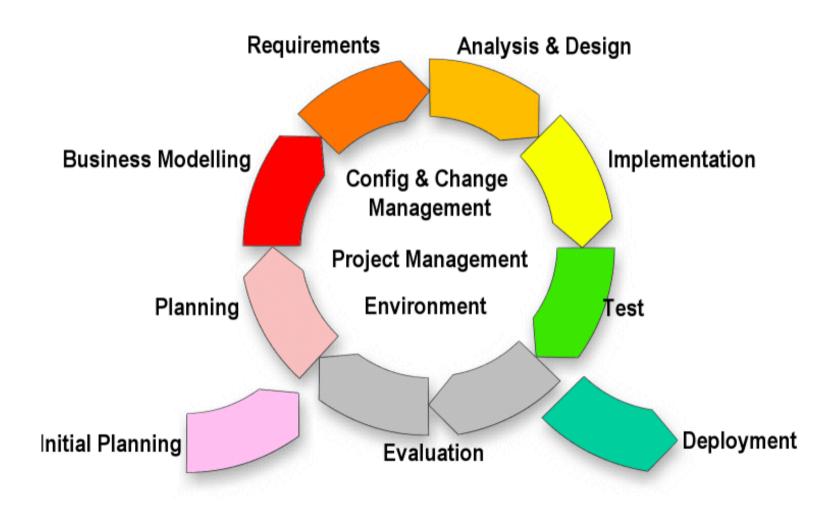
## Ciclo de vida





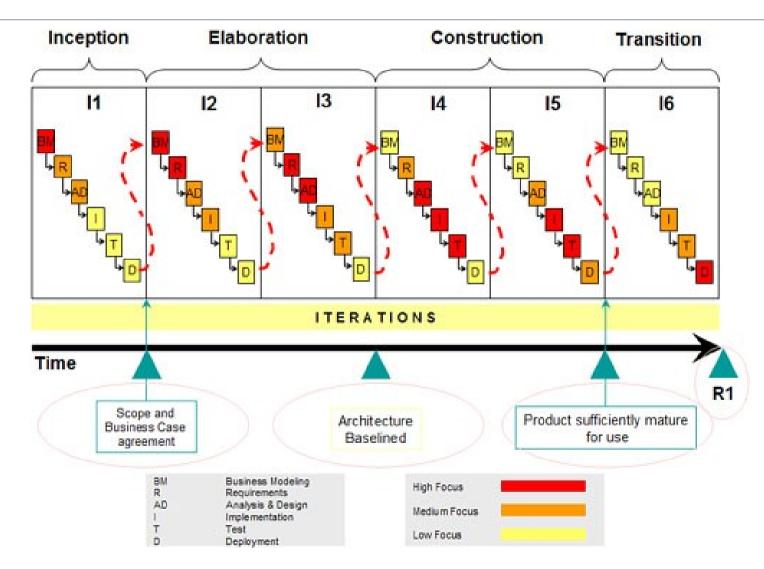
## Ciclo de vida





## Ciclo de vida





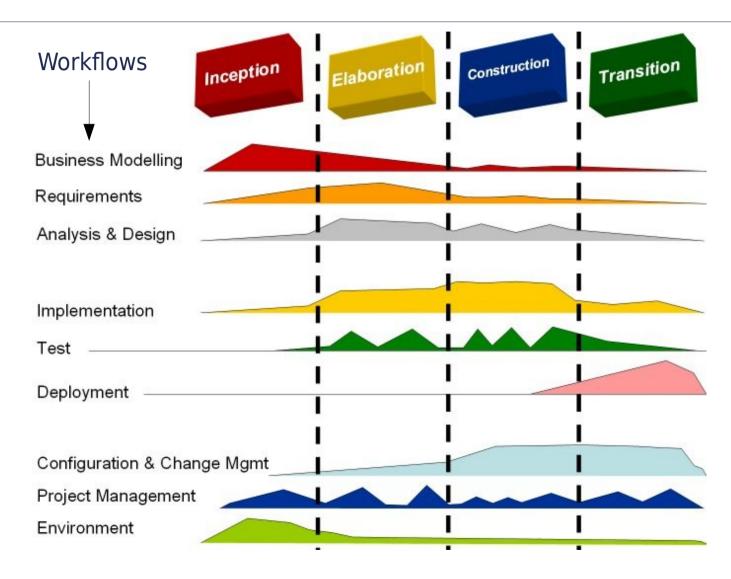
#### Workflows



- As atividades no RUP são chamadas de workflows
  - Existem seis workflows centrais
    - Modelagem de negócios
    - Requisitos
    - Análise e projeto
    - Implementação
    - Teste
    - Implantação
  - Três workflows de apoio
    - Gerenciamento de configuração e mudanças
    - Gerenciamento de projeto
    - Ambiente

## Workflows

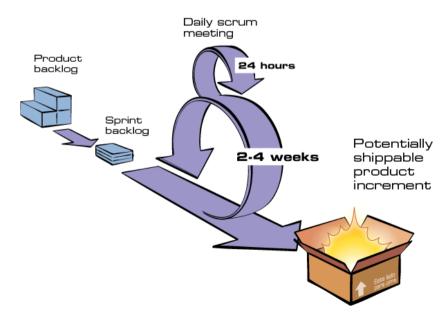




## RUP x Scrum

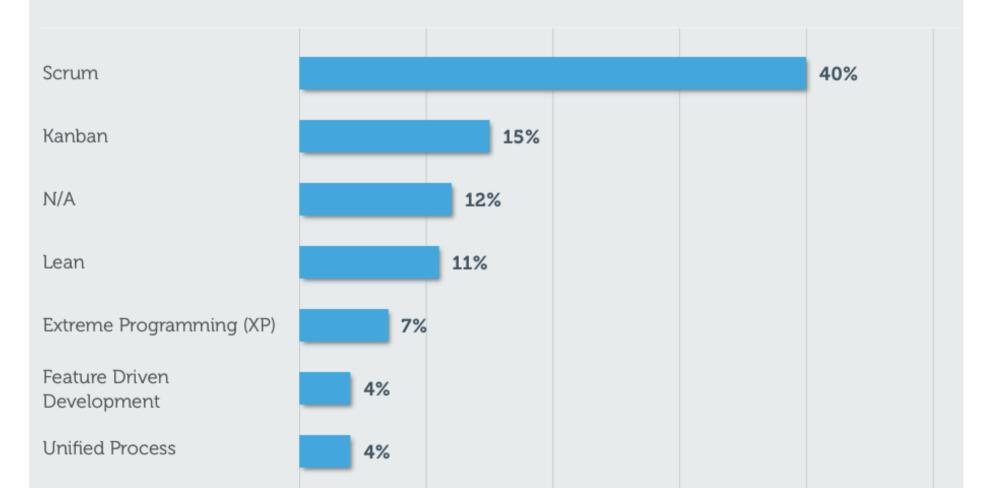






# What Agile approach is your organization using (select all that apply)?

Scrum led the way with 40%, followed by Kanban at 15% and Lean at 11%. 12% note that no Agile approach is used, indicating that a sizable percentage of our participants are not using Agile but want to learn more.



## Prescritivo x Adaptativo



#### < Prescritivo









Adaptativo >

## **Prescriptive vs Adaptive** processes

More prescriptive

**RUP** (120+)

Business use case realization Business use-case model

· Configuration audit findings

Configuration management

· Development-organization End-user support mateirla Glossary

Implementation model

Installation artifacts

Integration build plan

Issues list Iteration assessment

Programming guidelines

Quality assurance plan Reference architecture

Requirements attributes
 Requirements

 Risk management plan
 Software architecture document

Software development

· Software requirements

Supplementary business

Target organization assessment

Supplementary specification

· Test automation architecture Test cases

Test environment configuration

Test interface specification Test plan
 Test suite
 Tool guidelines

Use-case modeling guidelines Use-case realization

specification

Test ideas list

· Training materials

Use case package

Use-case storyboard

User-interface guidelines

User-interface prototype Vision

Use case model

Iteration plan

Release notes

Risk list

Manual styleguide

Business vision Change request

Deployment model

Deployment plan Design guidelines

Data model

 Design model Development case

- Architecture Reviewer Business Designer
- Business-Model Reviewer
   Business-Process Analyst Capsule Designer
- Change Control Manager
   Code Reviewer Configuration Manager
- Course Developer Database Designer Deployment Manager Design Reviewer
- Graphic Artist
- Integrator
   Process Engineer Project Manager Project Reviewe
- Requirements Reviewer
- Requirements Specifier
   Software Architect Stakeholder
- System Administrator System Analyst
- Technical Writer
   Test Analyst
- Test Designer Test Manager Tester
- Tool Specialist
   User-Interface Designer
   Architectural analysis
- Assess Viability of architectural proof-of-concept
- Capsule design
   Class design
   Construct architectural proof-
- of-concept Database design
- Describe distribution
   Describe the run-time architecture
- Design test packages and classes
- Develop design guidelines
   Develop programming
- guidelines Identify design elements
- Identify design mechanisms
   Incorporate design elements
   Prioritize use cases
- . Review the architecture Review the design
- · Structure the implementation model Subsystem design
- Use-case analysis Use-case design
- Analysis model Architectural proof-of-concept
   Bill of materials
- Business architecture document
- Business case Business glossary
- Business modeling guidelines
- Business rules Business use case
- Work order

(13)

XP

- Whole team
   Coding standard
  - Collective ownership Customer tests
  - Pair programming
     Refactoring Planning game Continuous integration
  - Simple design
     Sustainable pace
  - Metanhor Small releases

Soum (9)

- Scrum Master
   Product Owner Sprint planning
- Sprint planning meeting
   Daily Scrum
   Sprint review
   Product backlogt
- Sprint backlog BUrndown chart

Kanban (3)

· Visualize the workflow Measure and optimize lead Do Whatever

More adaptive

(0)

## O que o RUP não cobre?



 Gestão de pessoas: contratação, treinamento, acompanhamento

Gestão de orçamentos: definição, alocação, etc

 Gestão de contratos com fornecedores e clientes

## Vantagens



- Desenvolver iterativamente
  - Os maiores riscos são atacados primeiro, diminuindo as chances de fracasso do projeto
- Modelagem visual com UML
- Melhor controle sobre desenvolvimento:
  - Custos, prazos e níveis de qualidade desejados
  - Estimativa de prazos e custos com maior precisão
- Processo robusto e bem definido com a geração de artefatos importantes

## Desvantagens



 Complexo e trabalhoso para projetos de pequeno porte

- Necessita de um sério investimento em ferramentas de suporte
  - Normalmente fornecidas pela IBM
    - A IBM que mantém o RUP
- Exige treinamento intensivo e experiência da equipe
  - Muitas regras!

## Considerações finais



- Deve-se ter a consciência que os benefícios não virão de maneira imediata
  - É necessário:
    - Adquirir treinamento adequado
      - Apoio especializado para as equipes de desenvolvimento
    - Tempo para a absorção da metodologia
  - Indicado apenas para empresas e projetos de grande porte