

Levantamento, Análise e Gestão Requisitos

Aula 01



ENGENHARIA
DE REQUISITOS



Programadores {+ Inovadores ;}
www.x25.com.br

Fernando Anselmo

fernando.anselmo@x25.com.br

24 anos na área de Desenvolvimento e Coordenação
Empresas como Tecnisys, Politec, CF/OAB, FTI, FIA/USP...

13 Livros e artigos publicados

Instrutor e palestrante

Coordenador do DFJUG

Cargo Atual: Coordenador Técnico da X25 Informática

Página Oficial: <http://fernandoans.site50.net>



- Visão Geral
 - Contexto de "requisitos" dentro da engenharia de software
 - Perfil do analista e do gestor de requisitos
 - Importância e histórico sobre requisitos de software
 - Atividades envolvidas
 - Dificuldades e Negociações



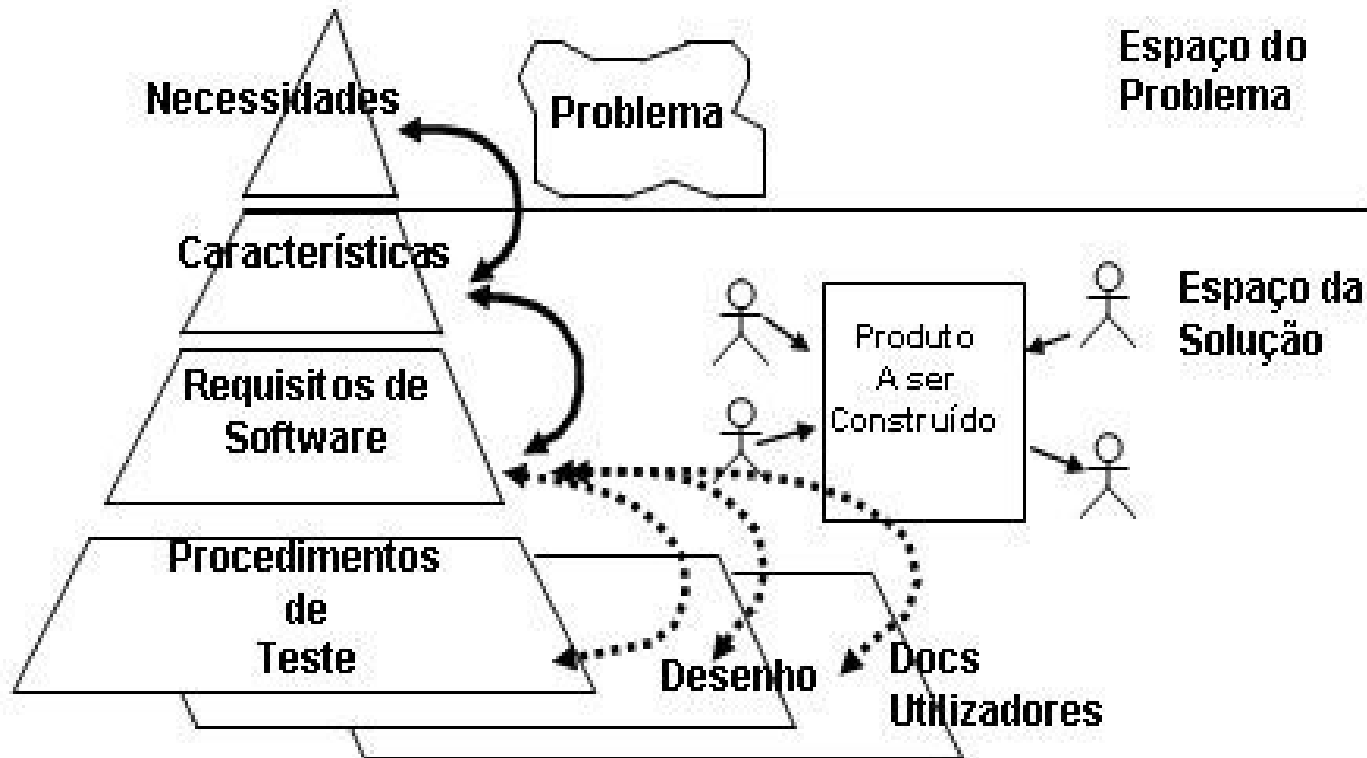
Visão Geral



Programadores (+ Inovadores ;)

www.x25.com.br

- A descrição do problema (acordada com o cliente)
- As várias partes interessadas (ou *stakeholders*) - perfis
- Os utilizadores (perfis) e sistemas na qual a solução interage (atores)
- As restrições do sistema (restrições ao sistema impostas externamente à equipe de desenvolvimento do projeto)



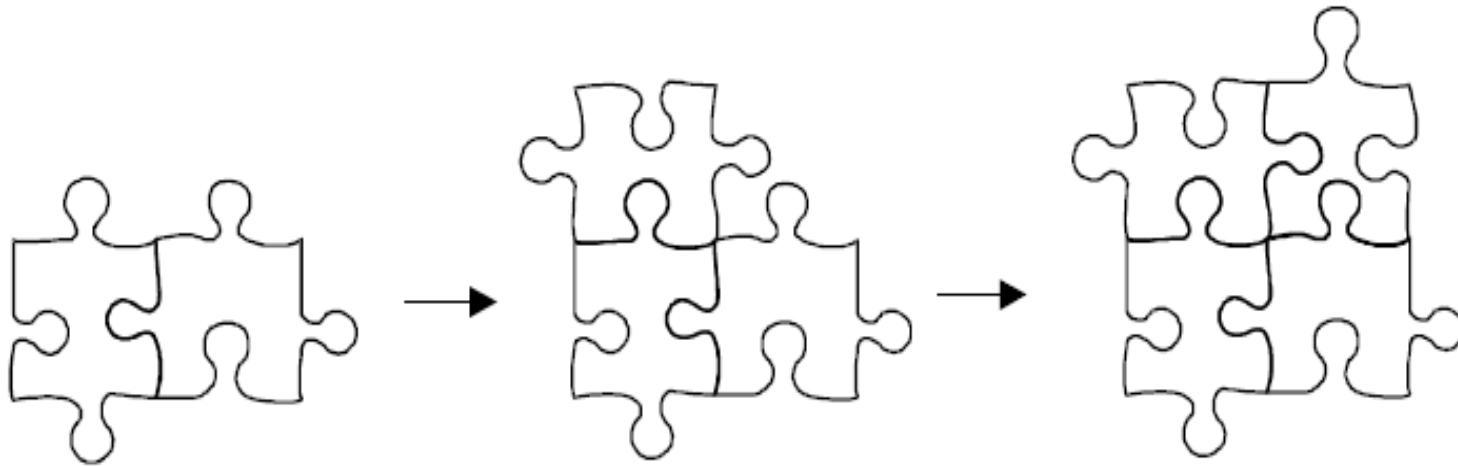
Processo Iterativo

Corresponde à ideia de “Melhorar (ou refinar) pouco a pouco” o sistema (iterações). Em cada iteração a equipe de desenvolvimento identifica e especifica os requisitos relevantes, cria um projeto utilizando a arquitetura escolhida como guia, implementa o projeto em componentes e verifica se esses componentes satisfazem os requisitos. Se uma iteração atinge os seus objetivos, o desenvolvimento prossegue com a próxima iteração, caso contrário a equipa deve rever as suas decisões e tentar uma nova abordagem.



Processo Incremental

Corresponde à ideia de “aumentar (alargar) pouco a pouco” o âmbito do sistema. Uma boa imagem para este atributo é a de uma mansão que foi construída por sucessivos incrementos a partir de uma primeira casa com apenas duas divisões.



Programadores {+ Inovadores ;}

www.x25.com.br

Diferenças do Processo Iterativo e Incremental

Iterativo



Incremental



Perfil do Analista e do Gestor de Requisitos



Programadores (+ Inovadores ;)

www.x25.com.br

Perfil do Analista e do Gestor de Requisitos

Conhecimentos: Engenharia de software; Análise orientada a objetos; Elicitação e gerenciamento de requisitos; Gestão de mudanças; Modelagem UML e de dados; métricas de software, desejável certificação em análise de pontos por função; Lógica de programação, desejável experiência com desenvolvimento de aplicações Client/Server, Windows Services, Webservices e Web; Banco de Dados.

Atitudes: organização e disciplina, excelência no atendimento a clientes, qualidade de trabalho, responsabilidade social, ética e integridade, iniciativa e comprometimento, comunicação, autodesenvolvimento.

Habilidades: relacionamento interpessoal, trabalho em equipe, capacidade de investigação análise e síntese, liderança, negociação, gestão de pessoas, planejamento, visão sistêmica.

Atividades a Serem Desenvolvidas: elicitação de requisitos; concepção, especificação, modelagem e teste de software, verificação de qualidade de artefatos e produtos de software, elaboração de diagnóstico e parecer técnico sobre sistemas, gestão do relacionamento entre fábrica de software e o cliente.



Objetivos da Análise de Requisitos

- Listar os sintomas dos problemas encontrados no Desenvolvimento de Software.
- Definir as Melhores Práticas.
- Descrever as Atividades para resolver problemas da Engenharia de Software para melhor prática.
- Descrever o Rational Unified Process® (RUP®) dentro do contexto das melhores práticas.

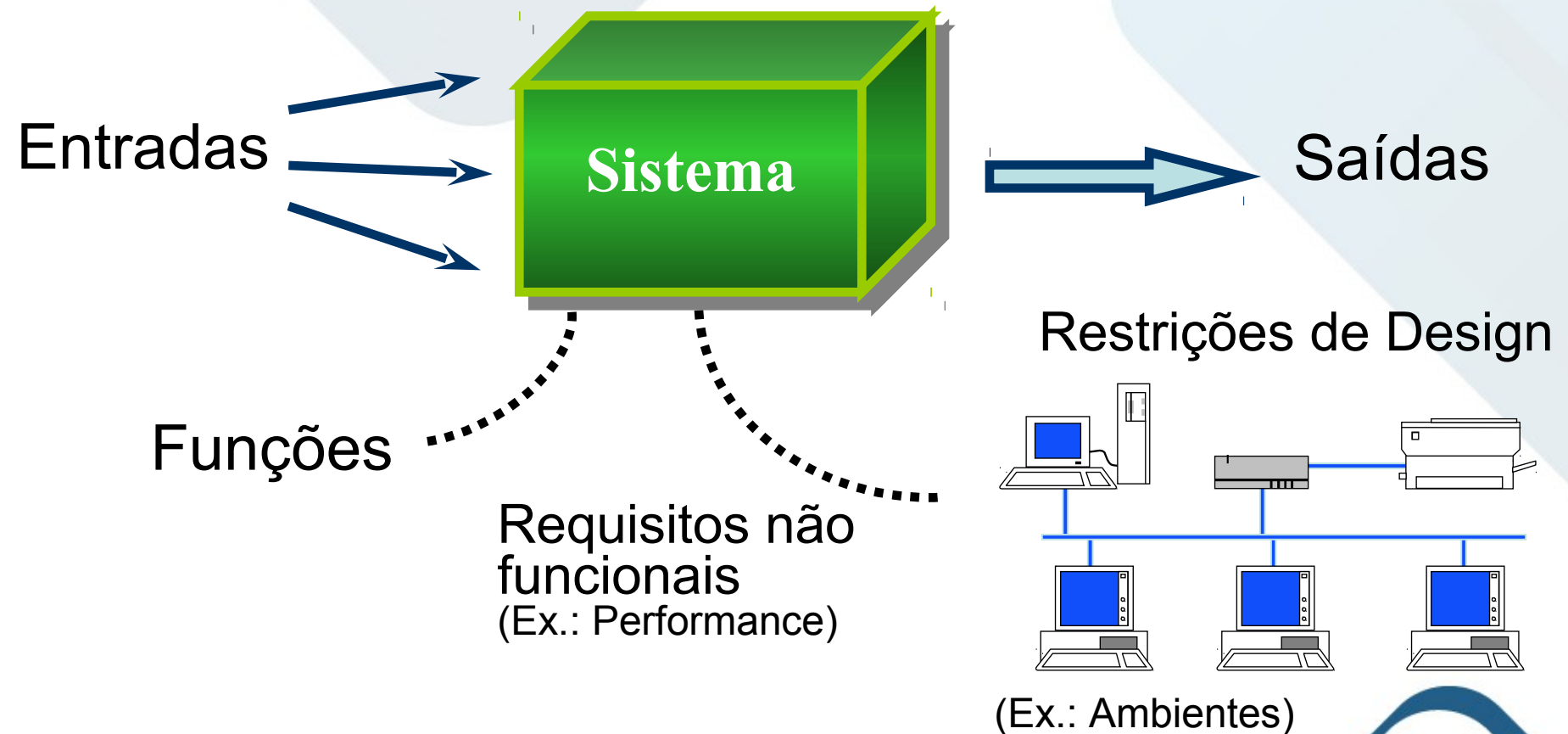


Sintomas dos Problemas do Desenvolvimento

- Necessidades de Negócio e Usuário não se encontram
- Muitas mudanças de requisitos
- Módulos não integram
- Difícil de manter
- Descoberta tardia das falhas
- Baixa qualidade e iteratividade com o usuário
- Baixa performance sob condições normais
- Esforço não coordenado da equipe
- Problemas de *build-and-release*



O que Requisitos de Software especificam?



Quando usar os Requisitos de Software?



Programadores (+ Inovadores ;)

www.x25.com.br

Mapeamento dos Sintomas para causas raiz

Sintomas

Necessidades erradas
Requisitos Mutantes
Módulos isolados
Difícil manter
Tarde demais
Baixa qualidade
Baixa performance
Equipe descoordenada
Problemas de *Build-e-release*



Causas Raiz

Requisitos insuficientes
Comunicação Ambígua
Arquiteturas fragmentadas
Complexidade desnecessária
Inconsistências
Testes insuficientes
Avaliação subjetiva
Desenvolvimento em cascata
Mudanças não controladas
Automação Insuficiente



Melhor Prática 1: Desenvolver Iterativamente

Melhores Práticas

Processo Prático

Desenvolver Iterativamente

Gerenciar Requisitos

Arquitetura de Componentes

Modelar Visualmente (UML)

Verificar Qualidade

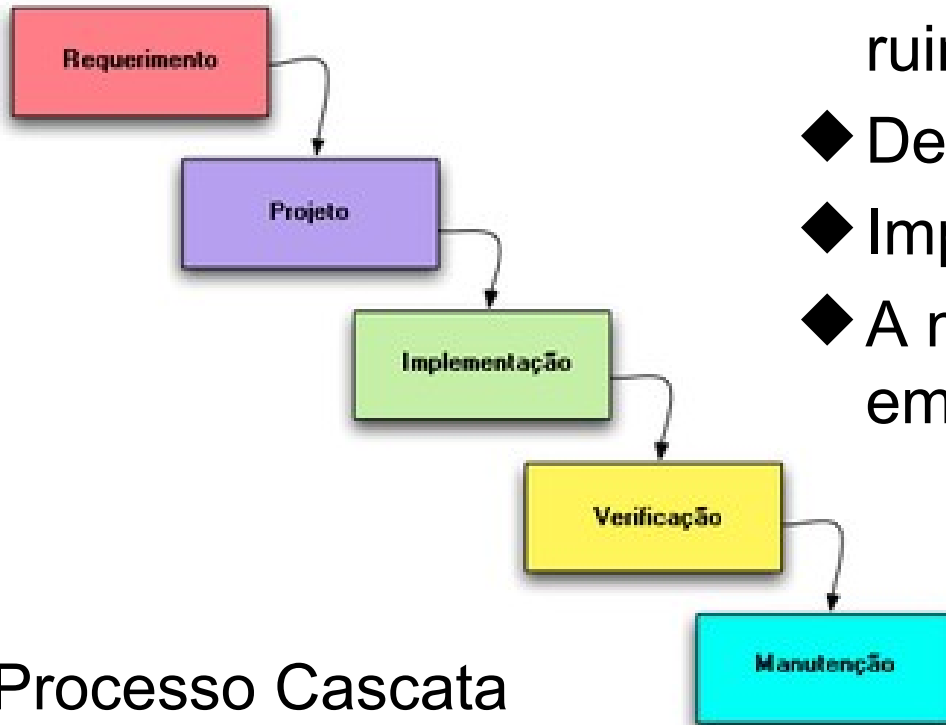
Gerenciar Mudanças



Programadores {+ Inovadores ;}

www.x25.com.br

Características do Desenvolvimento em Cascata



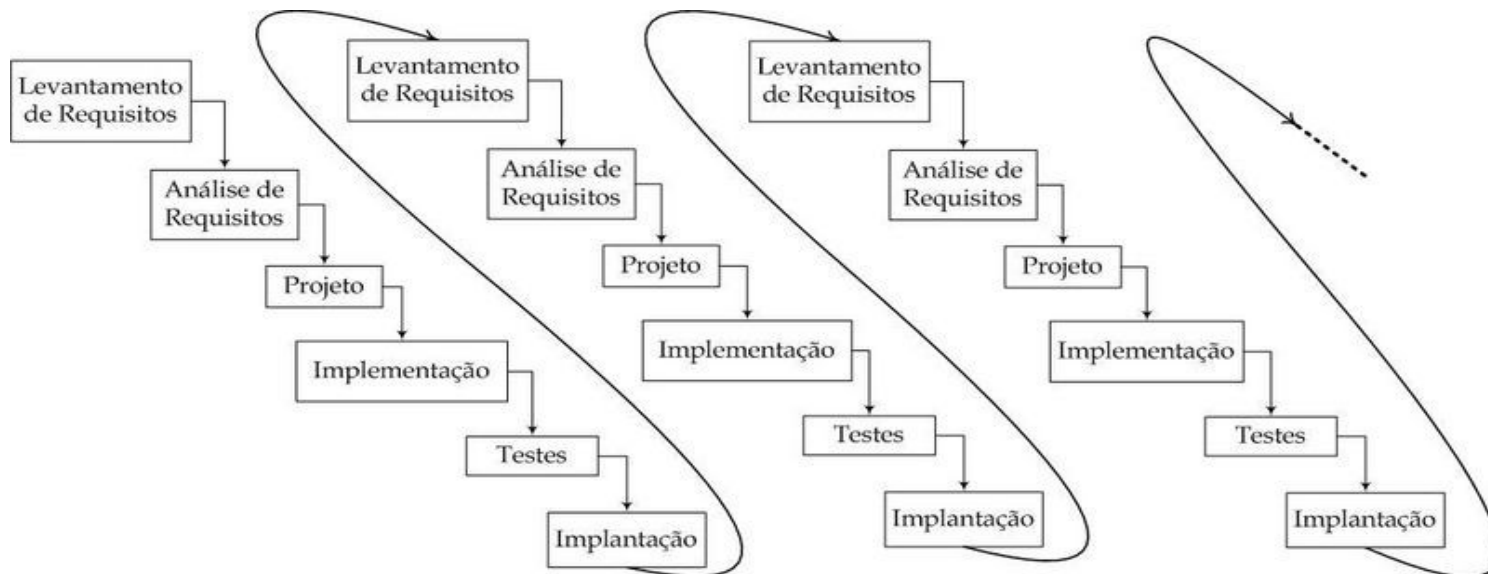
Processo Cascata

- ◆ Demora na confirmação e resolução de Riscos Críticos.
- ◆ Mensura progresso por produto-entregue que são indicadores ruins para avaliar progresso.
- ◆ Demora nas integrações e testes.
- ◆ Implantações no final.
- ◆ A maioria dos resultados ocorre em iterações não planejadas.

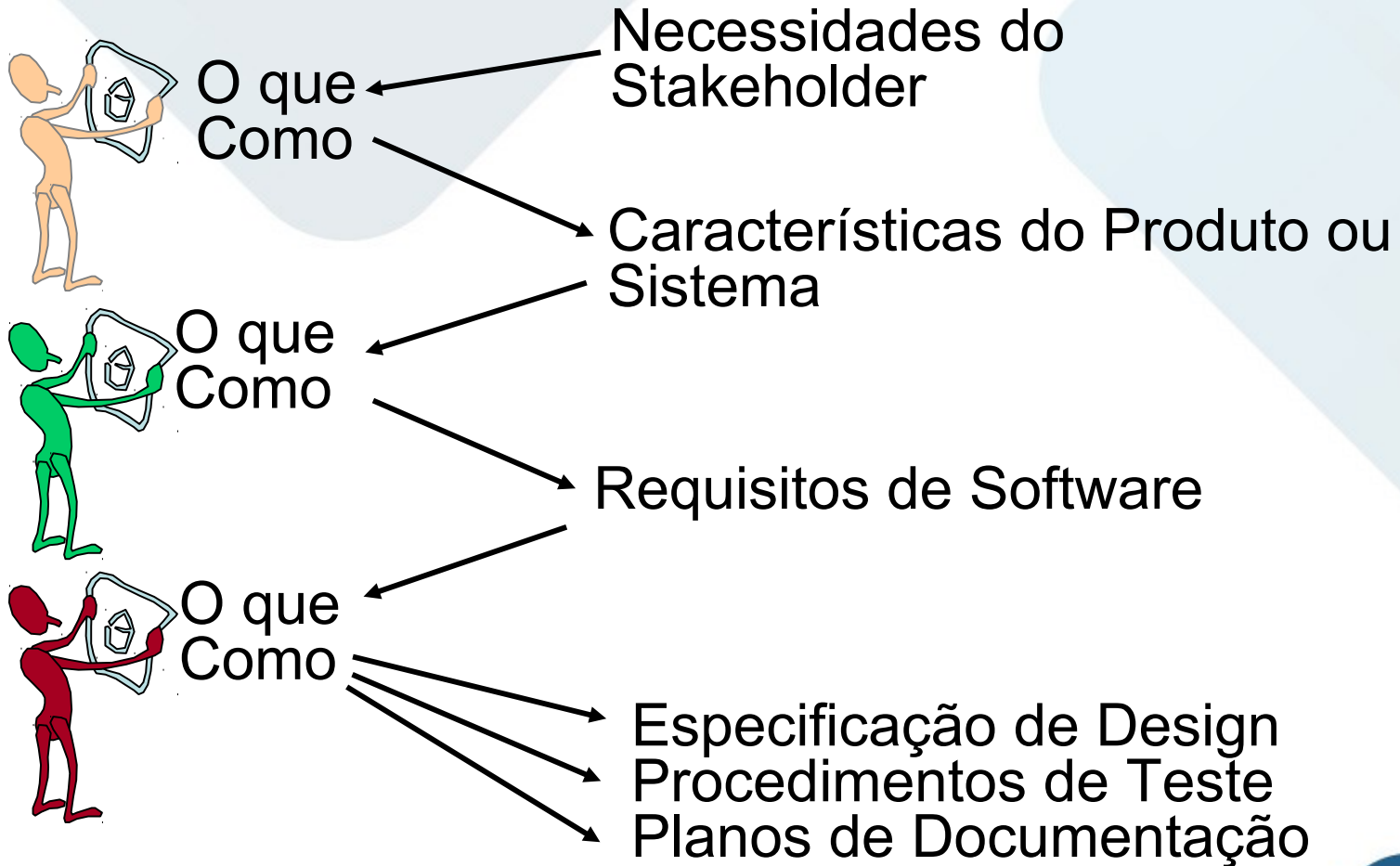


Características do Desenvolvimento Iterativo

- ◆ Resolve os maiores riscos antes que haja custos
- ◆ Facilita maior interação e feedback do usuário
- ◆ Faz teste e integração contínua
- ◆ Foca em marcos de projeto curtos e objetivos
- ◆ Torna possível a entrega parcial de executáveis

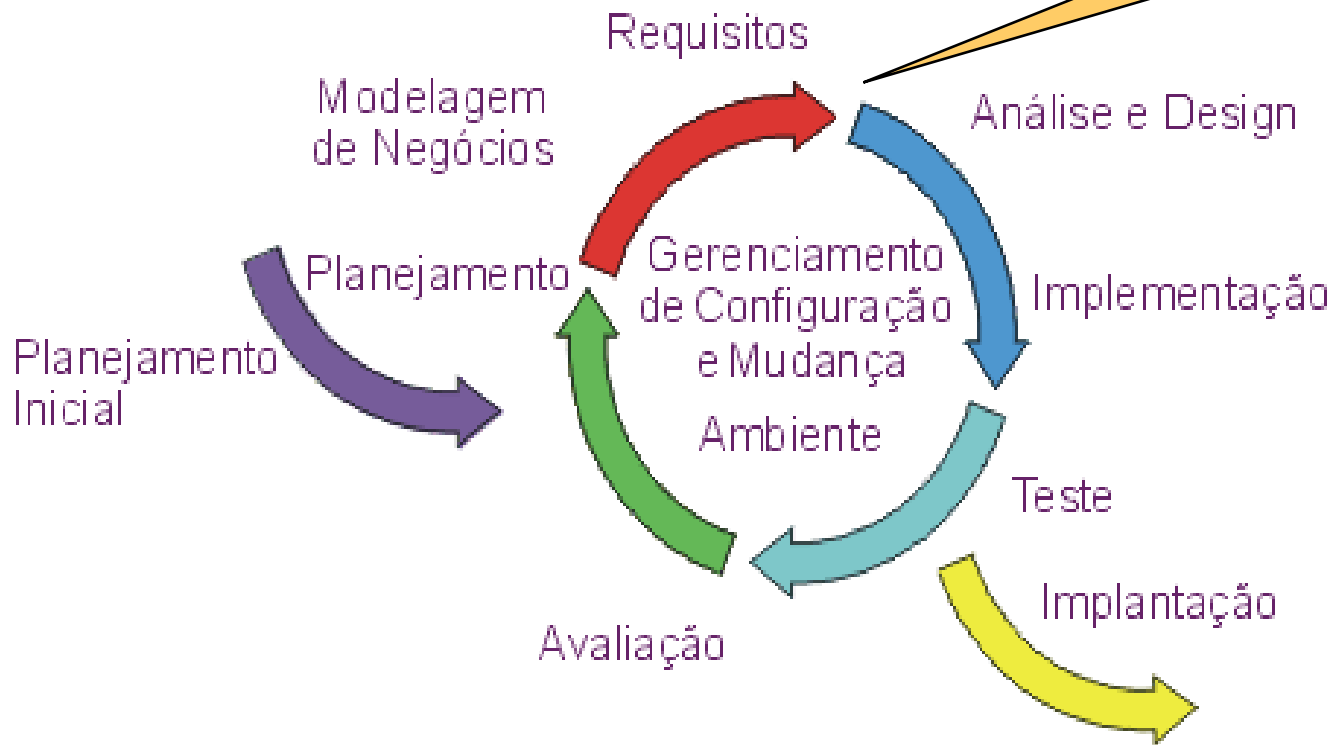


Requisitos existem em Vários Níveis

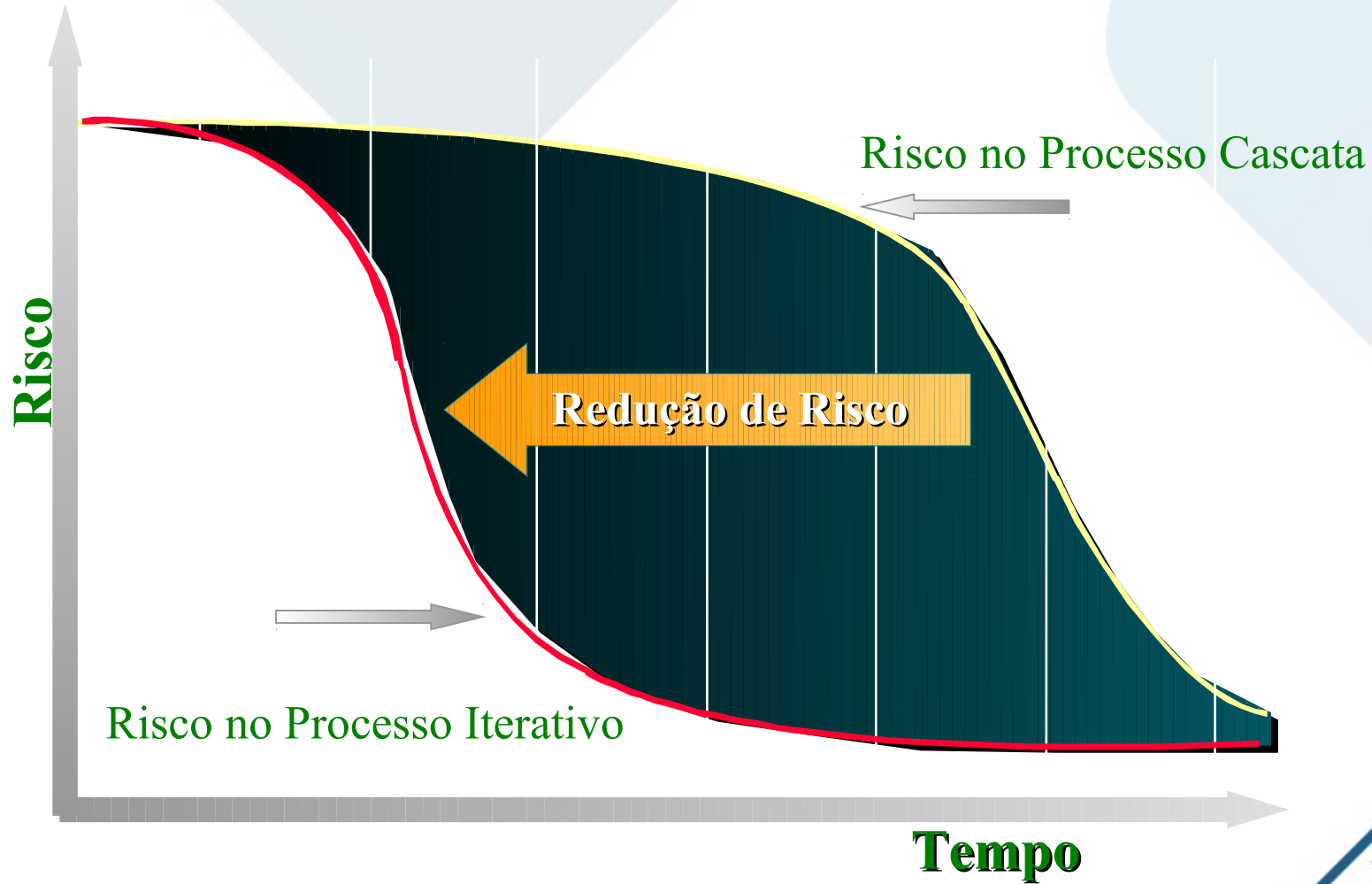


Desenvolvimento Iterativo produz um Executável

Cada iteração produz um executável (release).



Comportamento dos Riscos



Melhor Prática 2: Gerenciar Requisitos

Melhores Práticas

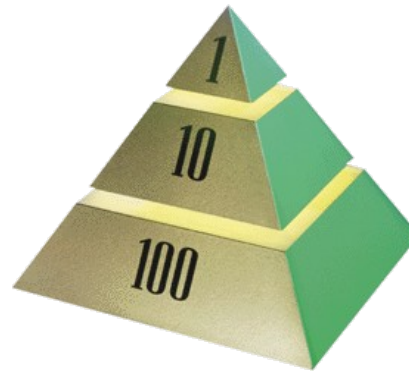
Processo Prático

Desenvolver Iterativamente
Gerenciar Requisitos
Arquitetura de Componentes
Modelar Visualmente (UML)
Verificar Qualidade
Gerenciar Mudanças



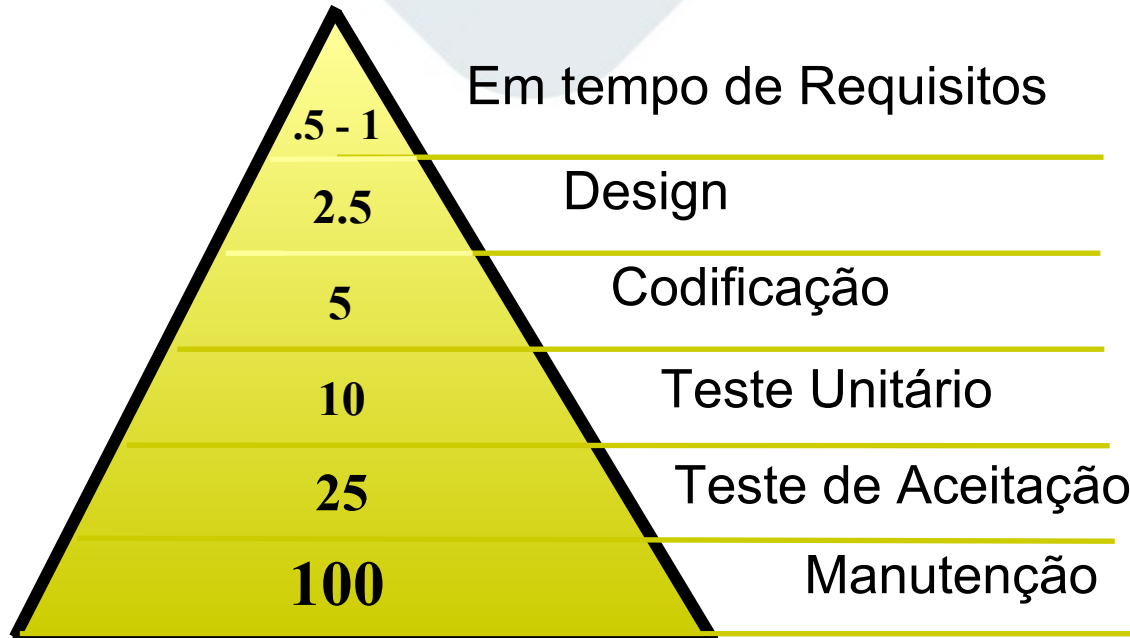
Gerenciamento de Requisitos

- ◆ Tenha certeza que está resolvendo o problema certo e entregando o sistema certo.
- ◆ Para ter uma ação completa:
 - ✓ Entenda o Problema.
 - ✓ Detalhe, organize, e documente os requisitos.
 - ✓ Gerencie as mudanças de requisitos do sistema.



Alto Custo de Requisitos Errados

A regra do 1-10-100



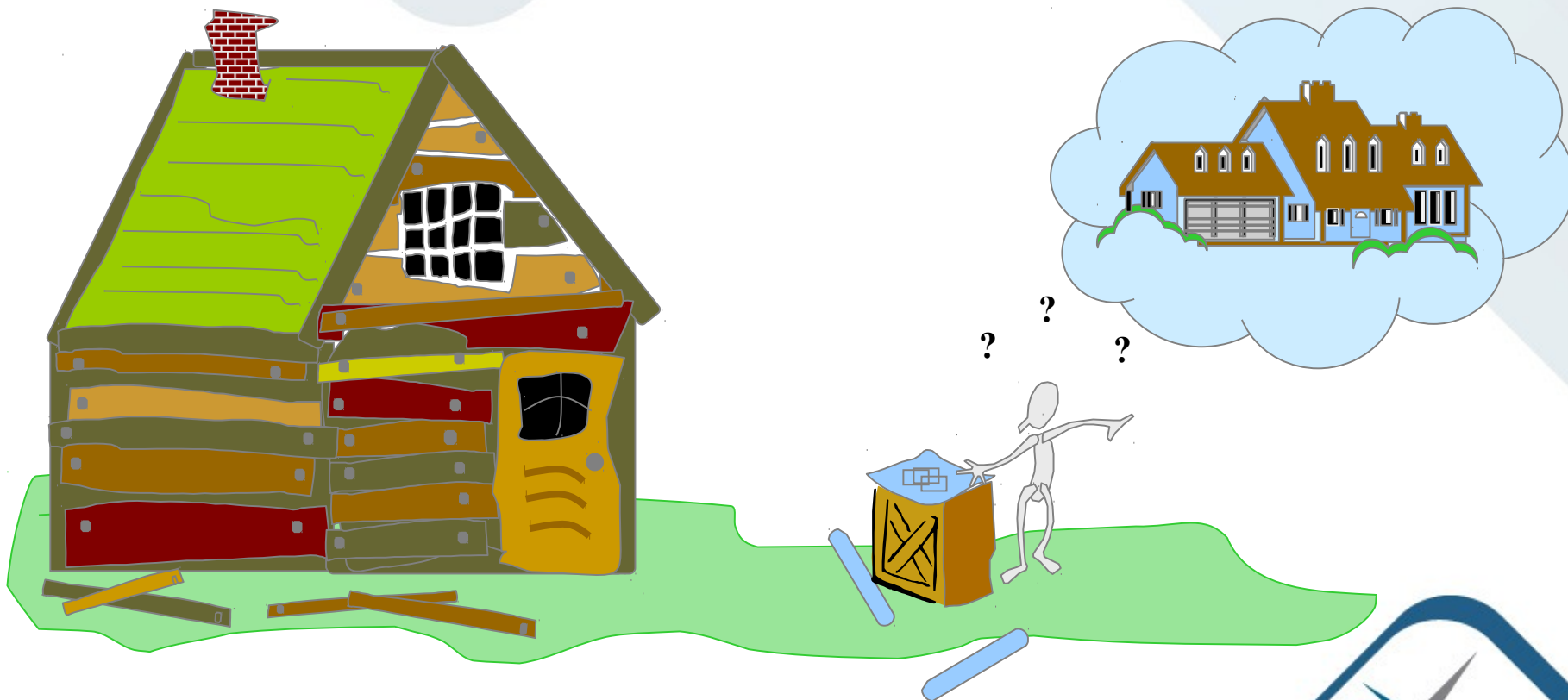
Custo relativo para reparar erros:
Quando Introduzidos X Quando reparados.



“All together, the results show as much as a 200:1 cost ratio between finding errors in the requirements and maintenance stages of the software lifecycle.”



Resultado pior quando a Qualidade é Baixa



Programadores {+ Inovadores ;}

www.x25.com.br

Melhor Prática 3: Arquitetura de Componentes

Melhores Práticas

Processo Prático

Desenvolver Iterativamente
Gerenciar Requisitos
Arquitetura de Componentes
Modelar Visualmente (UML)
Verificar Qualidade
Gerenciar Mudanças



Arquitetura Robusta Baseada em Componentes

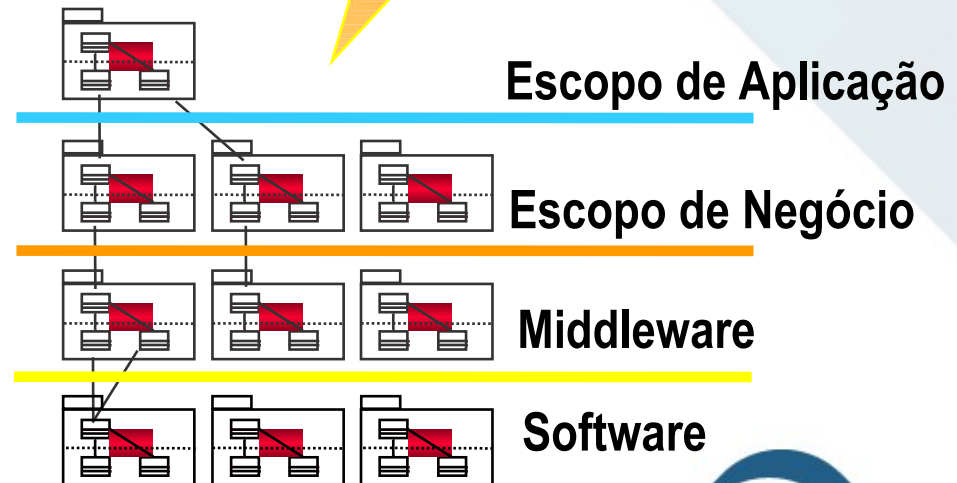
- Flexível
 - ✓ Atende aos requisitos atuais e futuros
 - ✓ Melhora a extensibilidade
 - ✓ Permite reuso
 - ✓ Encapsula as dependências do sistema
- Baseado em componentes
 - ✓ Reuso ou customização de componentes
 - ✓ Escolha de componentes comerciais disponíveis
 - ✓ Integra o sistema existente incrementalmente



Purpose of a Component-Based Architecture

- Base para Reuso
 - ✓ Componente
 - ✓ Arquitetura
- Base para Gerência de Projetos
 - ✓ Planejamento
 - ✓ Apoio
 - ✓ Entregas
- Controle Intelectual
 - ✓ Gerencia Complexidade
 - ✓ Mantém integridade

Arquitetura baseada em componentes com camadas



Melhor Prática 4: Modelar Visualmente (UML)

Melhores Práticas

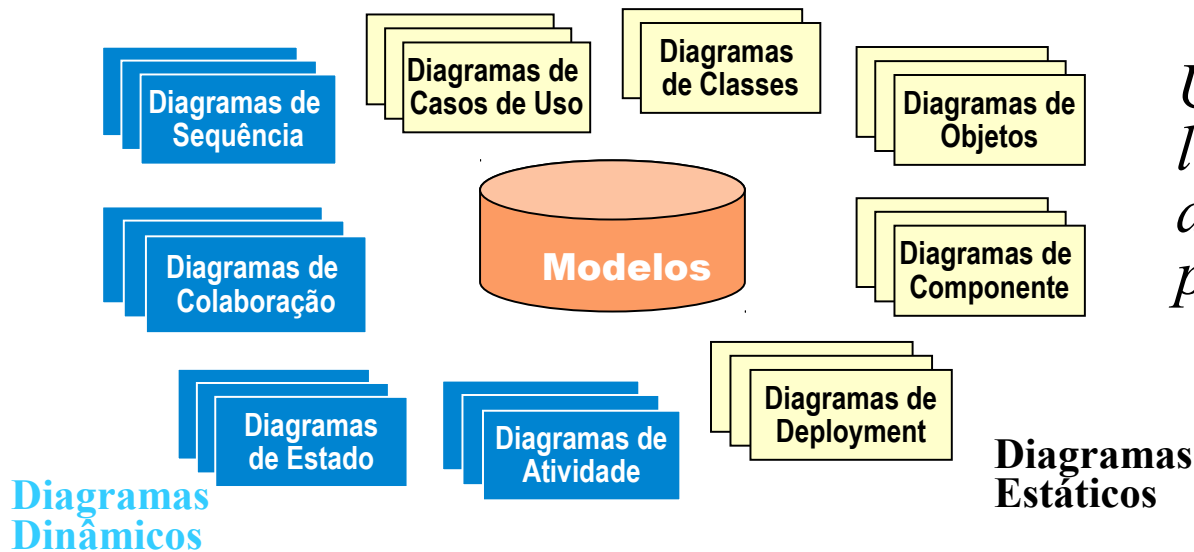
Processo Prático

Desenvolver Iterativamente
Gerenciar Requisitos
Arquitetura de Componentes
Modelar Visualmente (UML)
Verificar Qualidade
Gerenciar Mudanças



Por que Modelar Visualmente?

- Captura estrutura e comportamento
- Apresenta a colaboração entre os elementos
- Mantém design e implementação consistentes
- Esconde ou expõe detalhes conforme o caso
- Promove a comunicação não ambígua



UML, uma linguagem acessível a todos os praticantes.



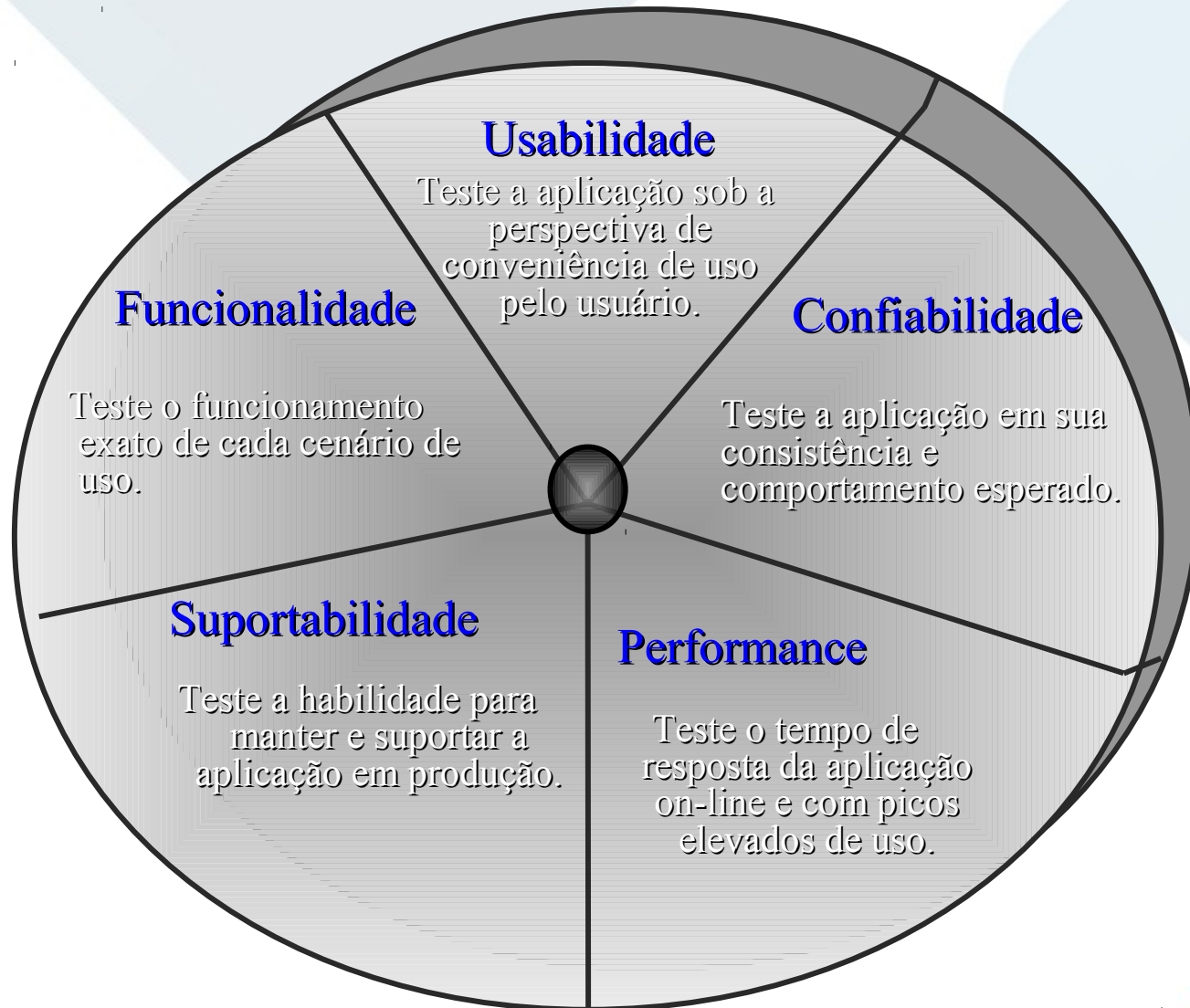
Melhor Prática 5: Verificar Qualidade

Melhores Práticas *Processo Prático*

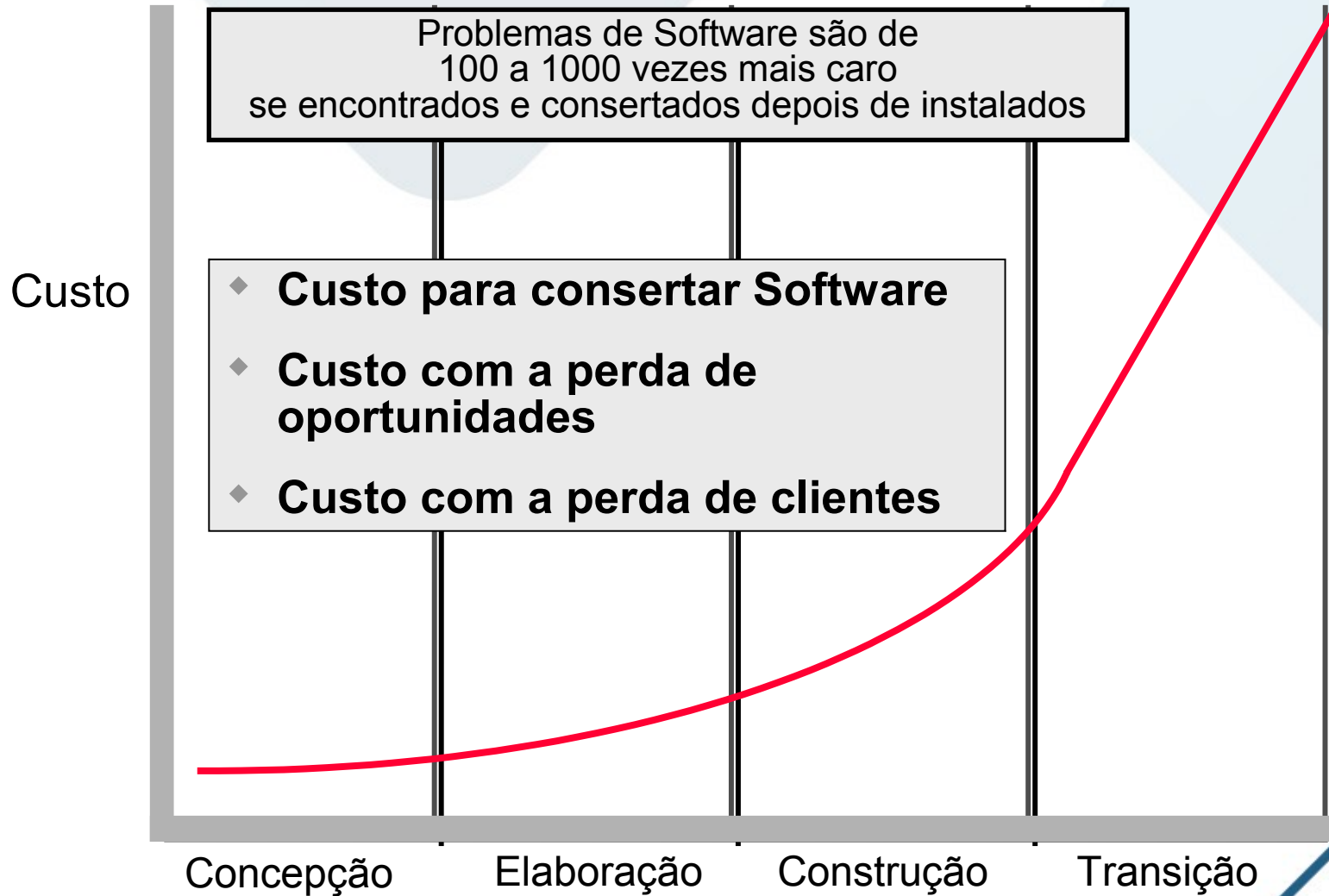
Desenvolver Iterativamente
Gerenciar Requisitos
Arquitetura de Componentes
Modelar Visualmente (UML)
Verificar Qualidade
Gerenciar Mudanças



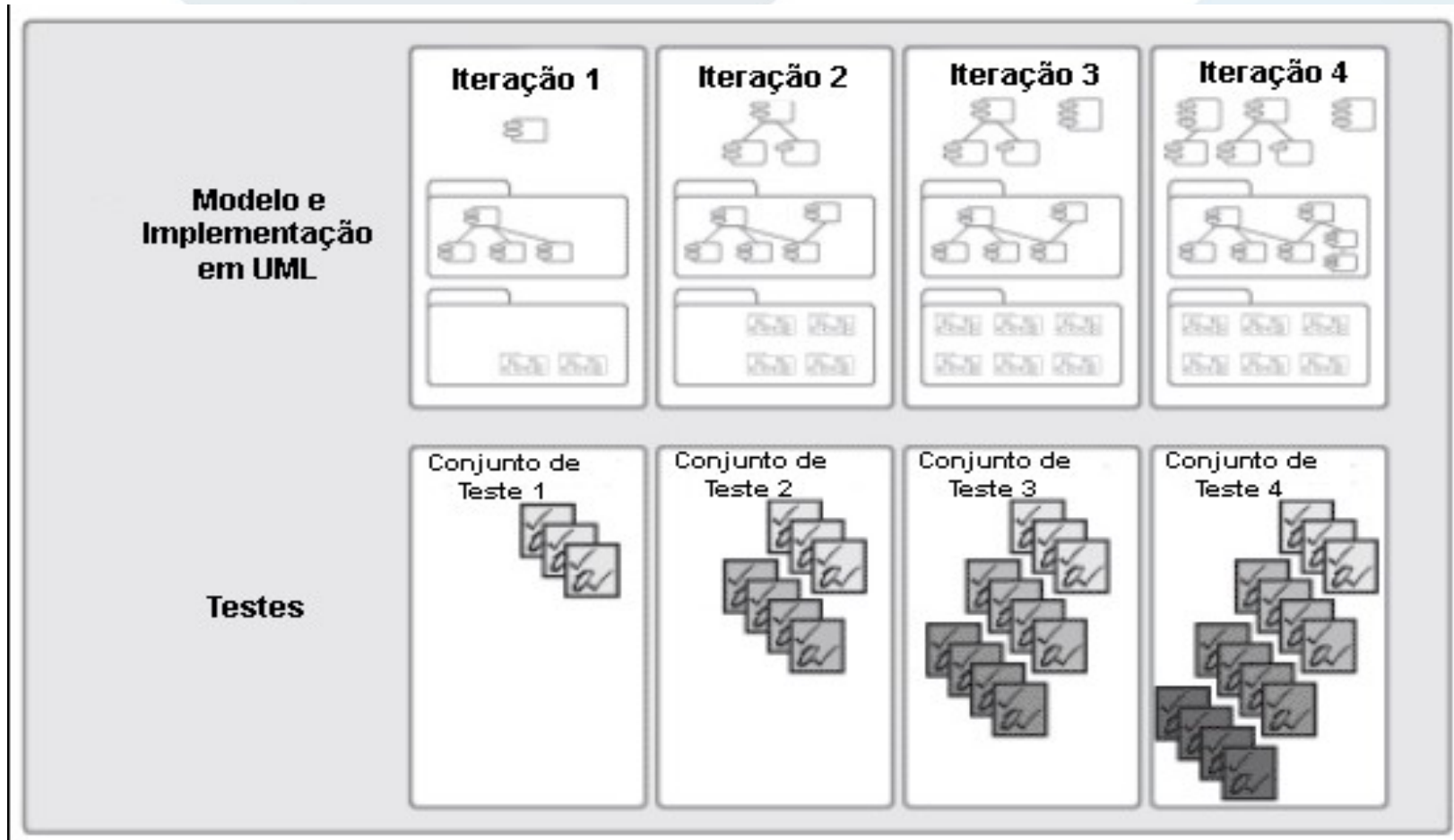
Dimensões de Teste de Qualidade



Verificação Contínua da Qualidade do Software



Teste cada Iteração



Melhor Prática 6: Gerenciar Mudanças

Melhores Práticas

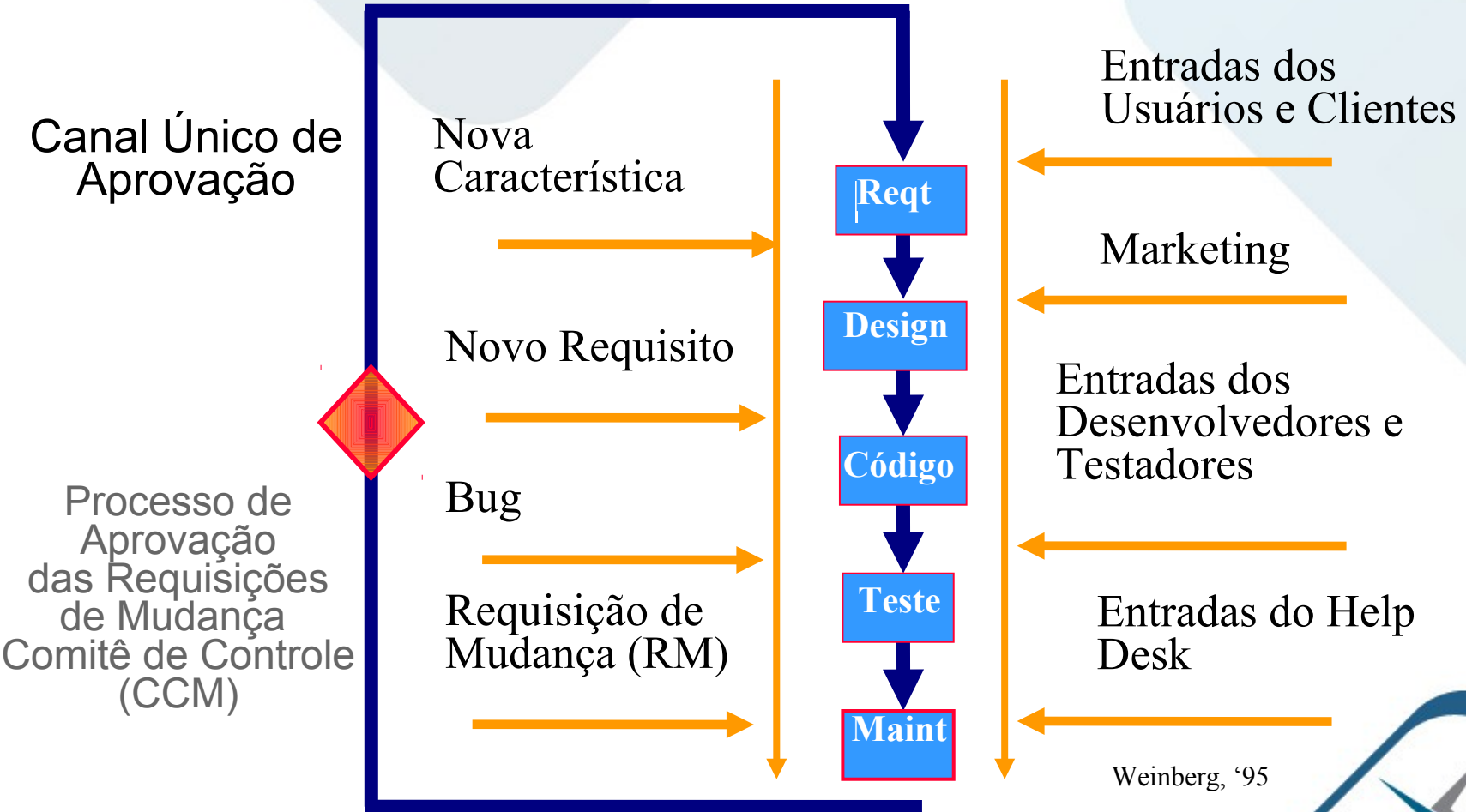
Processo Prático

Desenvolver Iterativamente
Gerenciar Requisitos
Arquitetura de Componentes
Modelar Visualmente (UML)
Verificar Qualidade
Gerenciar Mudanças



Gerenciamento das Requisições de Mudanças

Requisições de Mudanças vem de várias fontes durante o ciclo de vida do sistema.



Melhores Práticas Reforçam umas as outras

Melhores Práticas

Desenvolver Iterativamente

Gerenciar Requisitos

Arquitetura de Componentes

Modelar Visualmente (UML)

Verificar Qualidade

Gerenciar Mudanças

Envolvimento dos usuários no desenvolvimento dos requisitos

Validação antecipada das decisões arquiteturais

Ataca complexidade de Design / Implementação incrementalmente

Mensura qualidade frequentemente

Desenvolvimento incremental de baselines



Rational Unified Process



Melhores Práticas *Processo Prático*

Desenvolver Iterativamente

Gerenciar Requisitos

Utilizar Arquitetura de Componentes

Modelar Visualmente (UML)

Verificação Contínua da Qualidade

Gerenciar Mudanças



Programadores {+ Inovadores ;}

www.x25.com.br

Gerenciamento de Requisito Efetivo

- Manter um claro estabelecimento dos requisitos requer:
 - Boa qualidade dos requisitos
 - Atributos aplicáveis para cada tipo de requisito
 - Rastreamento para outros requisitos e outros artefatos do projeto


O OBJETIVO é entregar produtos de **qualidade**
No tempo e no orçamento que atendam
As reais necessidades do usuário



Um Processo de Definição Baseado em Equipe

O processo define **QUEM** faz **O QUE**, **QUANDO** e **COMO** para atingir um determinado objetivo


Novo ou alterado
Requisitos



Processo de
Requisitos



Novo ou alterado
Sistema



Programadores {+ Inovadores ;}

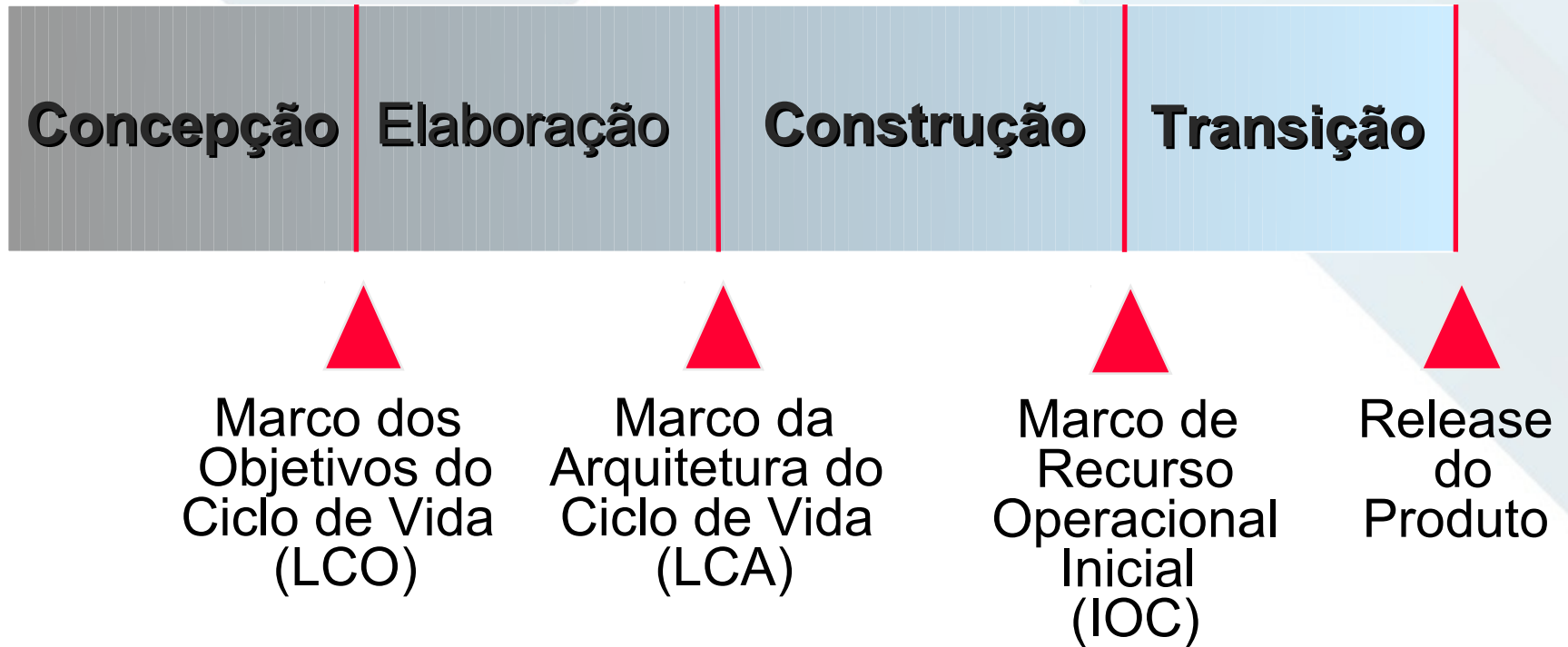
www.x25.com.br

Estrutura do Processo – Fases do Ciclo de Vida

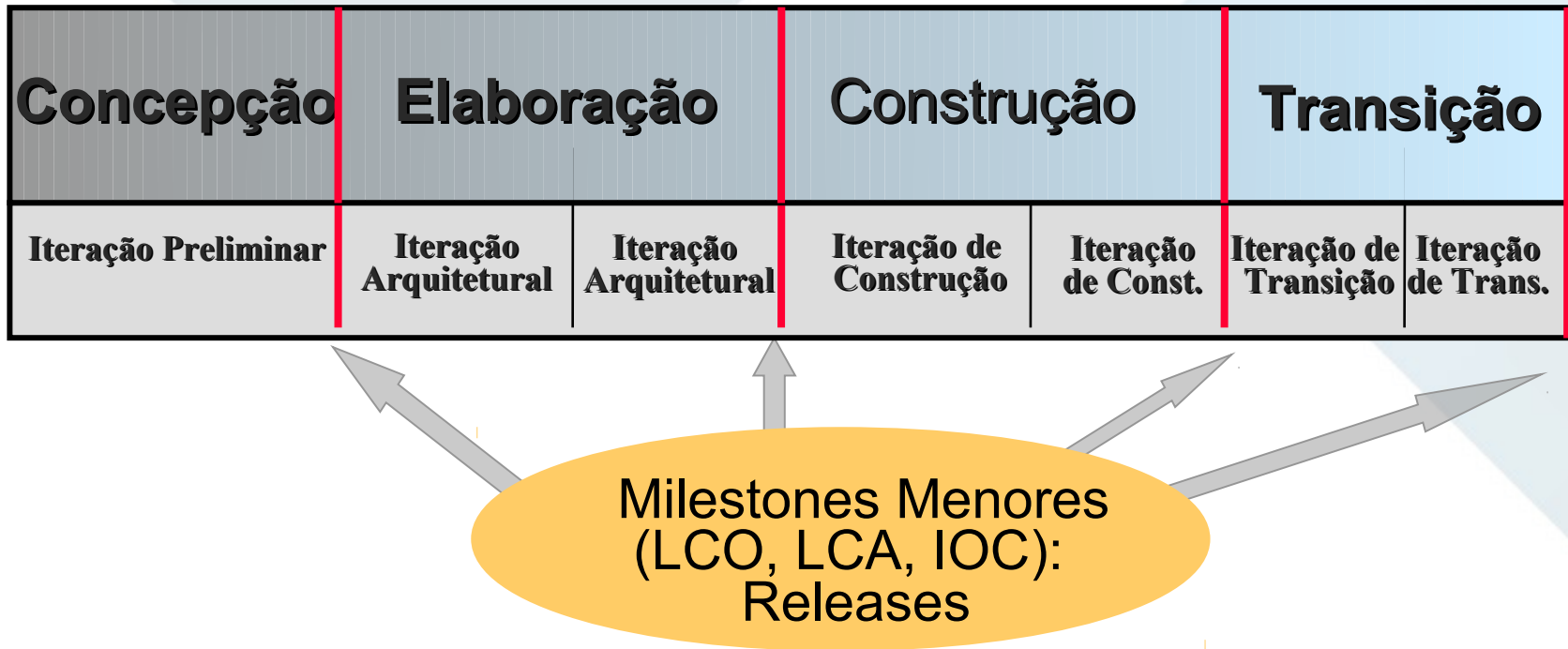
Iniciação	Elaboração	Construção	Transição
Lidar com os riscos relacionados ao processo de negócio. (parte financeira do projeto)	Lidar com os riscos relacionados aos riscos técnicos do projeto.	Lidar com os riscos relacionados a “ter a massa de trabalho pronta.”	Lidar com os riscos relacionados a logística de implantação da aplicação para sua base de usuários.



Fronteiras das Fases marcam os Maiores Milestones



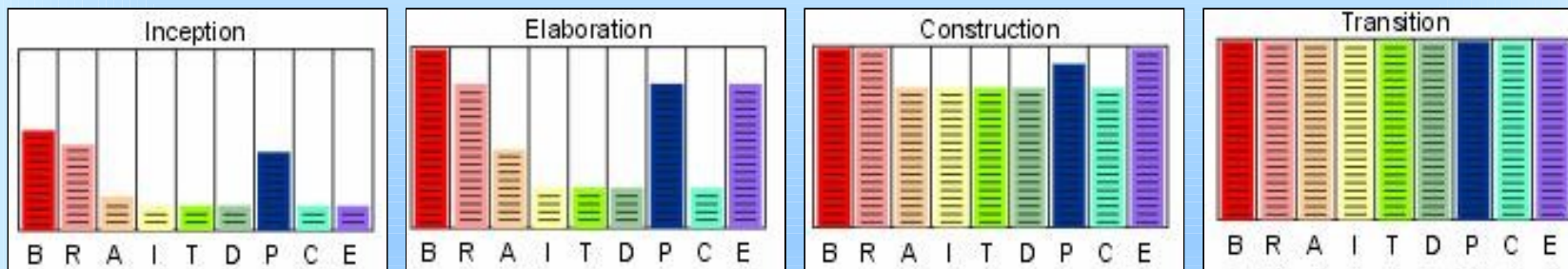
Iterações e Fases









Uma **iteração** é uma sequência distinta de atividades baseadas em um plano existente e critérios de avaliação, resultando na liberação de um executável (interno ou externo).

Artefatos completos do Ciclo de Vida do Projeto

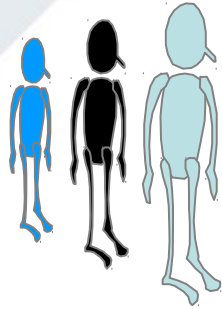
Artefatos do Projeto são produzidos iterativamente



 B	: Business Modeling Set	 D	: Deployment Set
 R	: Requirements Set	 P	: Project Management Set
 A	: Analysis & Design Set	 C	: Configuration & Change Management Set
 I	: Implementation Set	 E	: Environment Set
 T	: Test Set		



On Time and On Budget

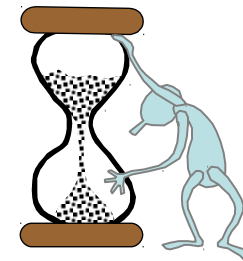
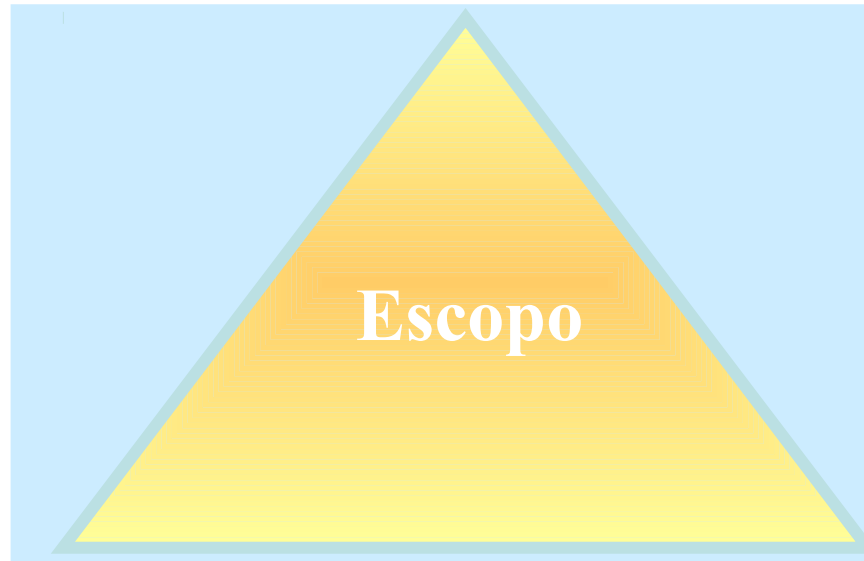


Recursos

Quanto de trabalho podemos fazer?



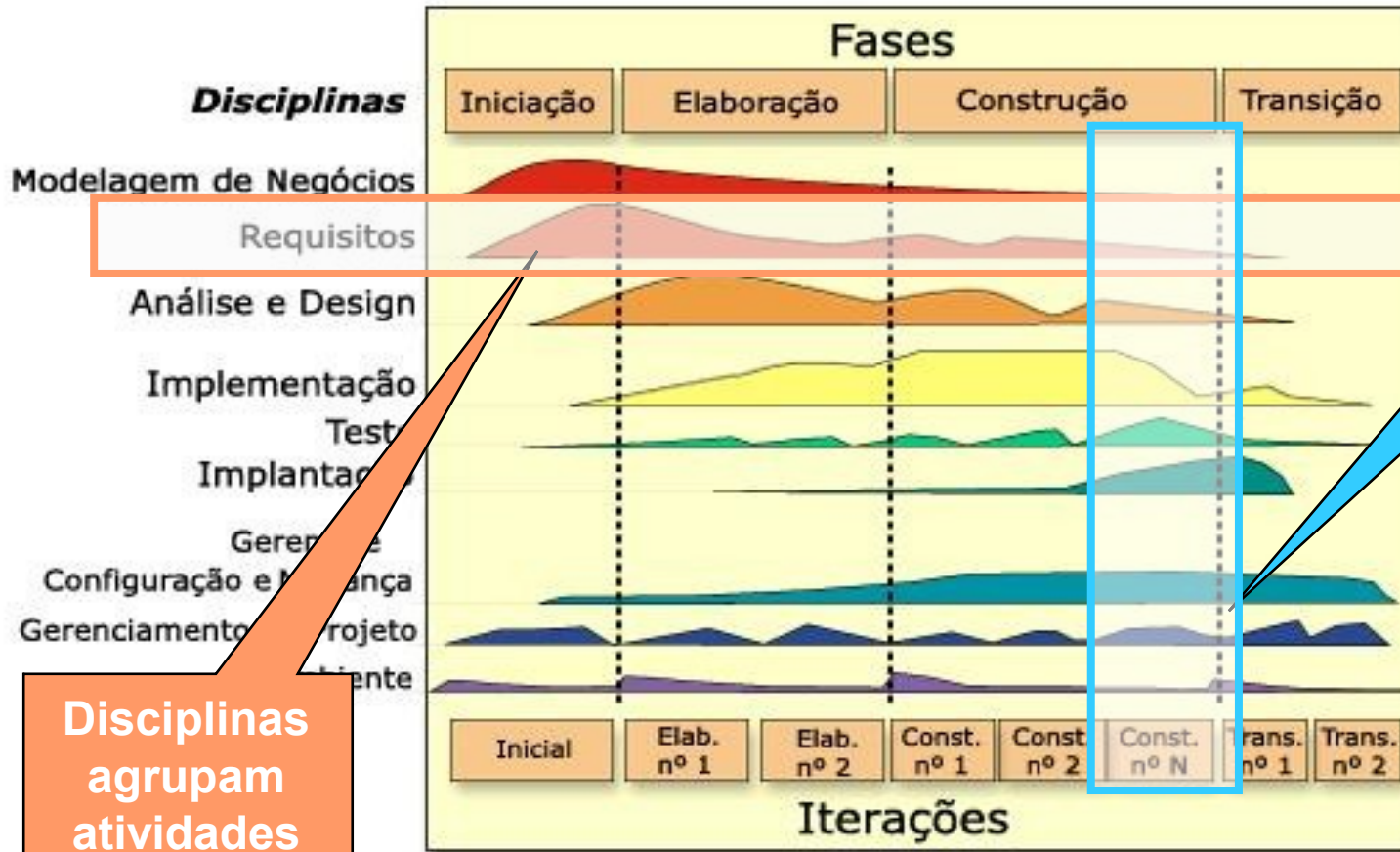
Orçamento



Tempo



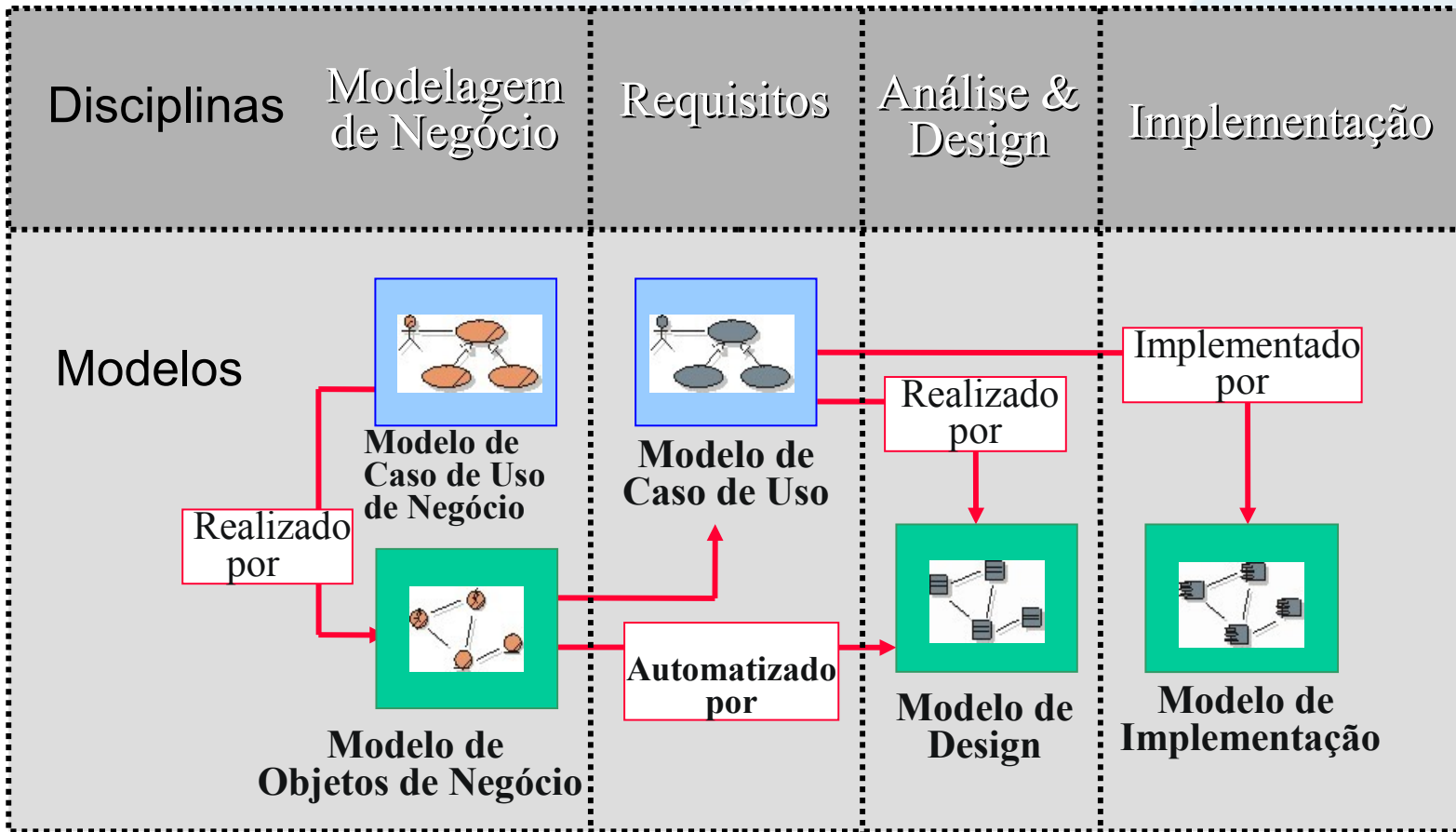
Colocando tudo junto: Uma abordagem Iterativa



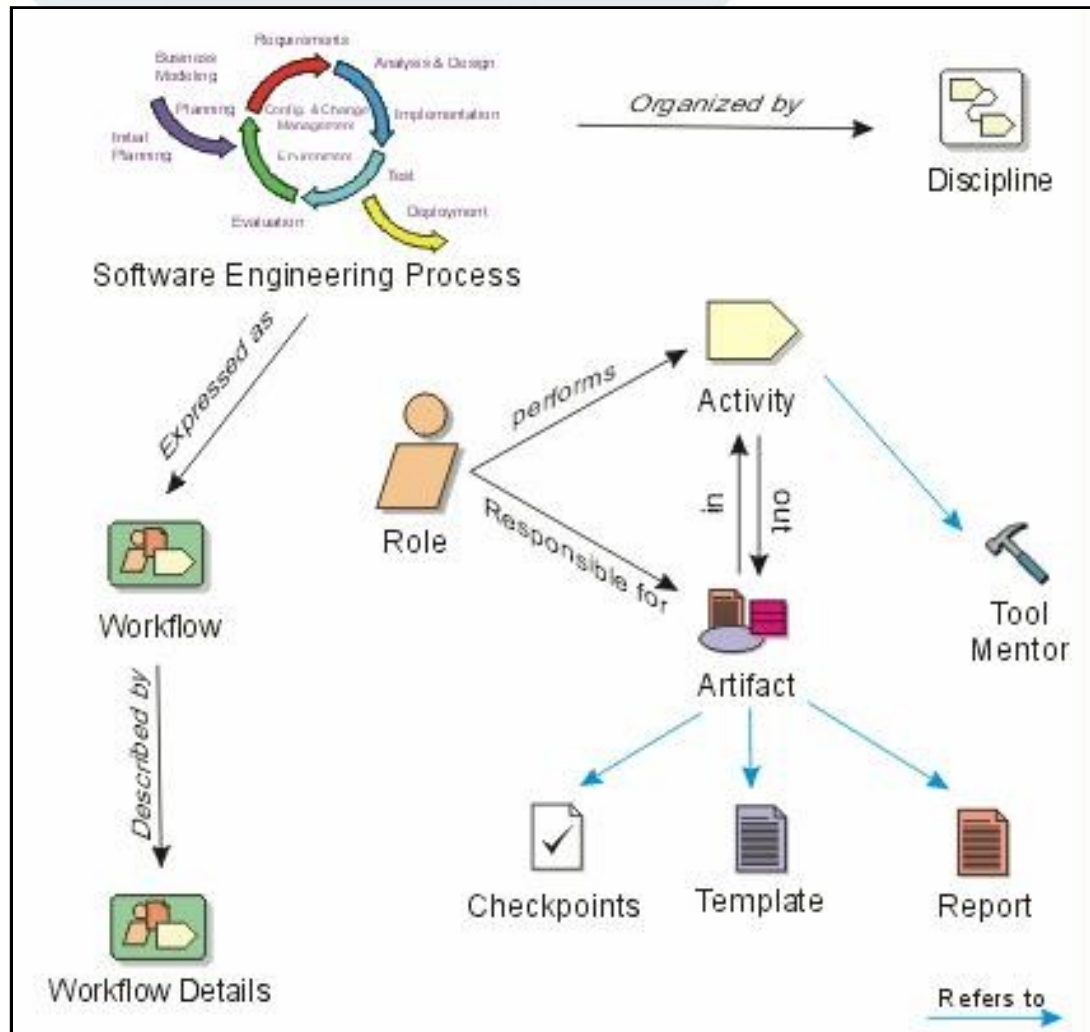
Em uma iteração, você passa por todas as disciplinas.

Disciplinas agrupam atividades logicamente

Disciplinas produzem modelos



Visão Geral dos conceitos do RUP



- As melhores práticas são guias da Engenharia de Software para atacar as causas dos problemas do Desenvolvimento de Software.
- As melhores práticas reforçam umas as outras.
- O Processo guia a equipe **de quem, o que, quando e como** fazer as atividades
- **Rational Unified Process** é uma forma de colocar em ação as Melhores Práticas.



Dúvidas? Agradecimentos

Home Page

<http://fernandoans.site50.net>

Blog

<http://fernandoanselmo.blogspot.com>

X25 Home Page

<http://www.x25.com.br>



Fernando Anselmo

fernando.anselmo@x25.com.br



Programadores {+ Inovadores ;}

www.x25.com.br