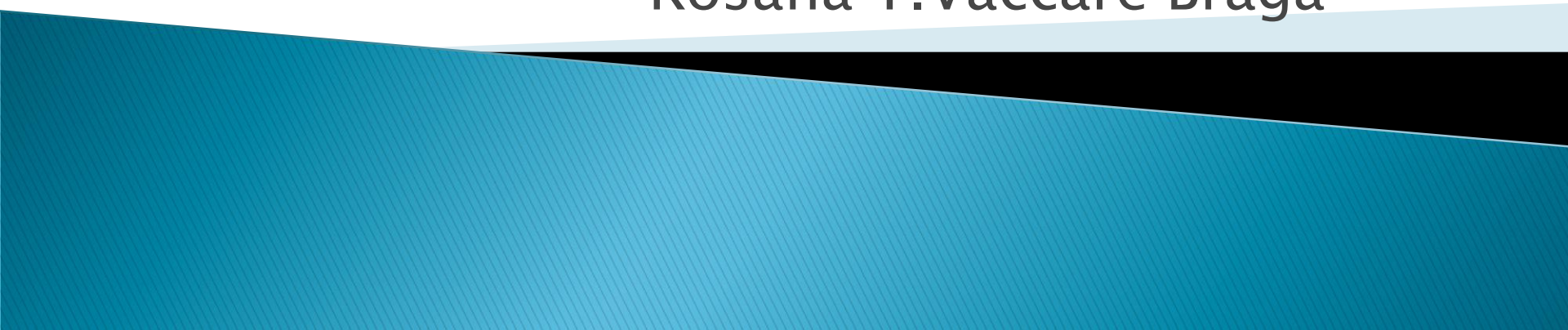
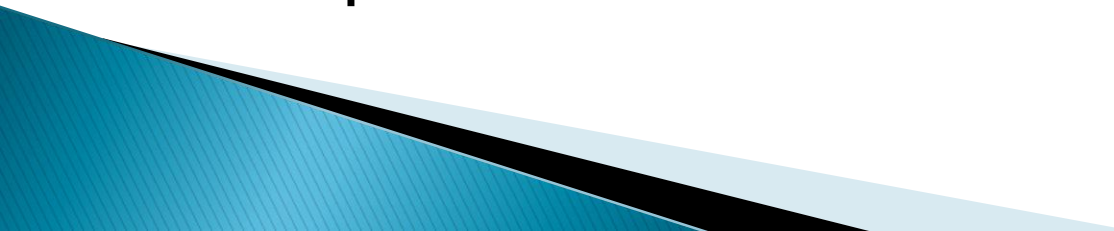


Introdução ao Reúso de Software

Rosana T.Vaccare Braga



Introdução

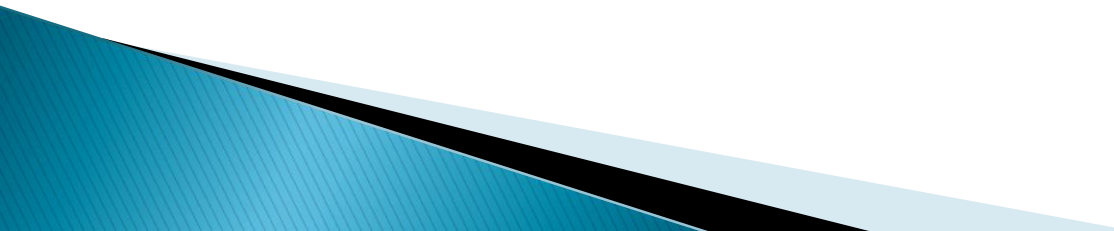
- ▶ Engenharia de Software baseada em Reúso é uma abordagem de desenvolvimento que tenta maximizar a reutilização de software existente.
 - ▶ Tem sido um objetivo da Engenharia de Software desde 1968.
 - ▶ Ocorre normalmente de forma ad hoc
 - ▶ Deve-se fazer com que o reúso ocorra de forma sistemática e com suporte computacional.
- 

Introdução

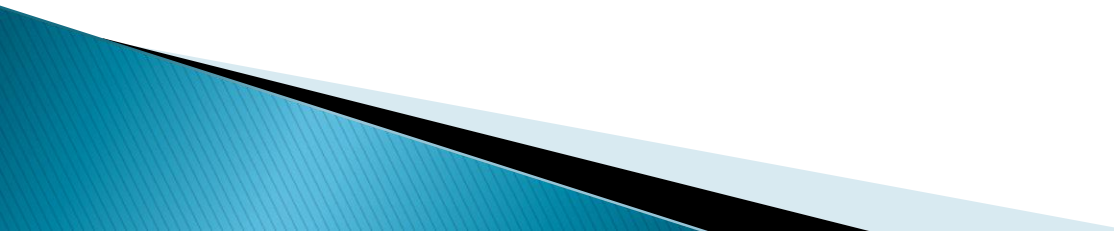
► Reúso de Software:

- Anos 70: módulos e sub-rotinas
- Anos 80: classes e geradores de aplicação
- Anos 90: análise de domínio, componentes, padrões, frameworks
- Anos 00: aspectos, serviços, MDA
- Anos 10: SaaS, PaaS, SoS, IoT...

Benefícios do Reúso

- ▶ Desenvolvimento acelerado (eficiência)
 - ▶ Conformidade com padrões (Ex. GUI)
 - ▶ Confiança aumentada
 - ▶ Risco de processo reduzido
 - ▶ Uso eficiente de especialista (evitar retrabalho)
- 

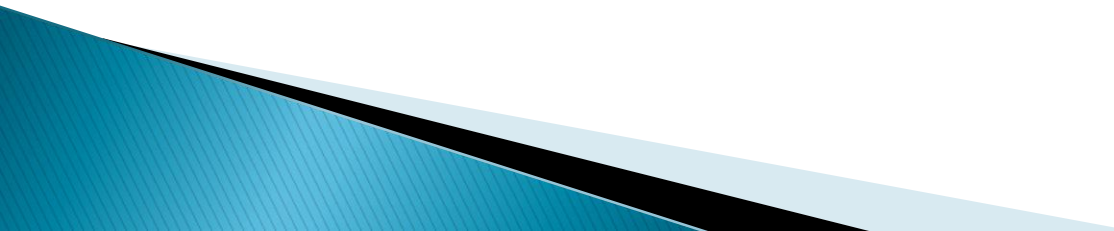
Possíveis problemas com o reúso

- ▶ Custos de manutenção aumentados (se não tiver acesso ao código fonte)
 - ▶ Falta de apoio de ferramentas
 - ▶ Síndrome do não-inventado-aqui
 - ▶ Criação e manutenção de uma biblioteca de componentes (ou ativos)
 - ▶ Procura, compreensão e adaptação de componentes reusáveis.
- 

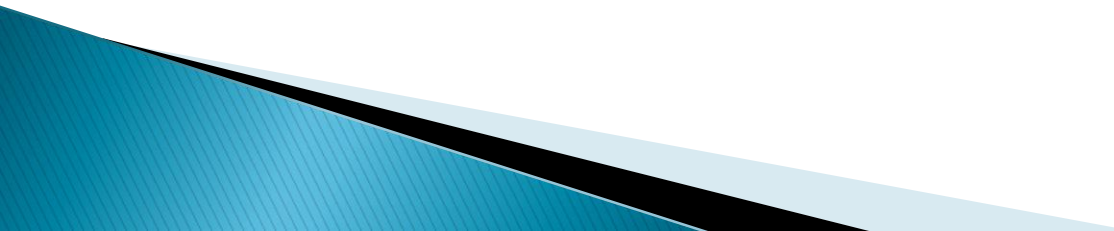
Reúso de Software

- ▶ Pode ocorrer em diversos níveis:
 - Código (componentes ou unidades de software)
 - Projeto Detalhado
 - Arquitetura
 - Modelos de Análise
 - “Idéias” ou soluções para problemas recorrentes
- ▶ Quanto mais alto o nível, geralmente mais altos são os ganhos.

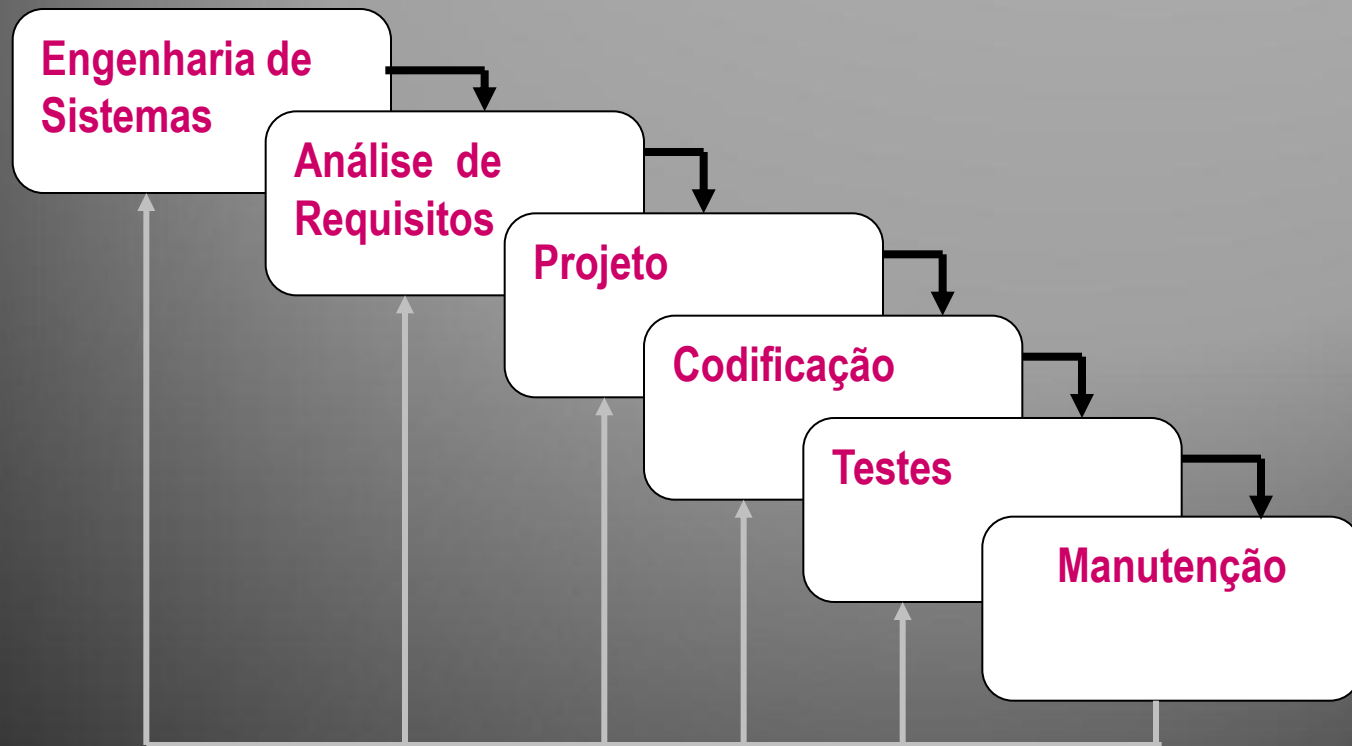
Reúso de Software

- ▶ Oportunístico (ad hoc) ou Planejado?
 - ▶ Exemplo de reúso oportunístico: cortar e colar.
 - ▶ Exemplo de reúso planejado: reúso de um framework desenvolvido previamente.
 - ▶ A tecnologia determina o tipo de reúso? (Pense em componentes, por exemplo?)
- 

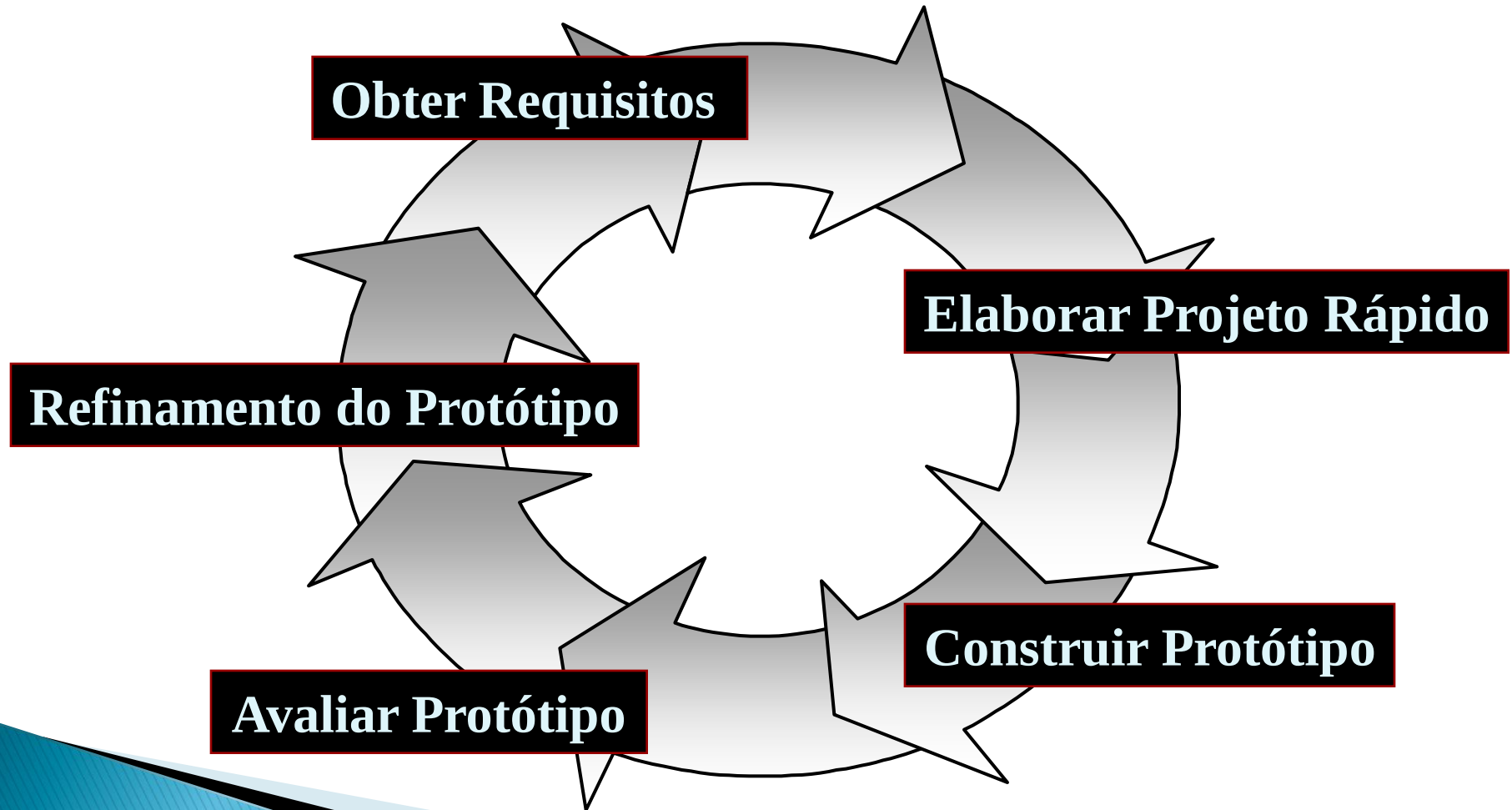
Reúso de Software

- ▶ O processo de desenvolvimento tem que levar em consideração o reúso de software
 - ▶ Pense nos modelos de processo que você conhece: cascata, prototipação, incremental, Processo Unificado, Scrum, etc.
 - ▶ Onde o reúso se encaixa?
- 

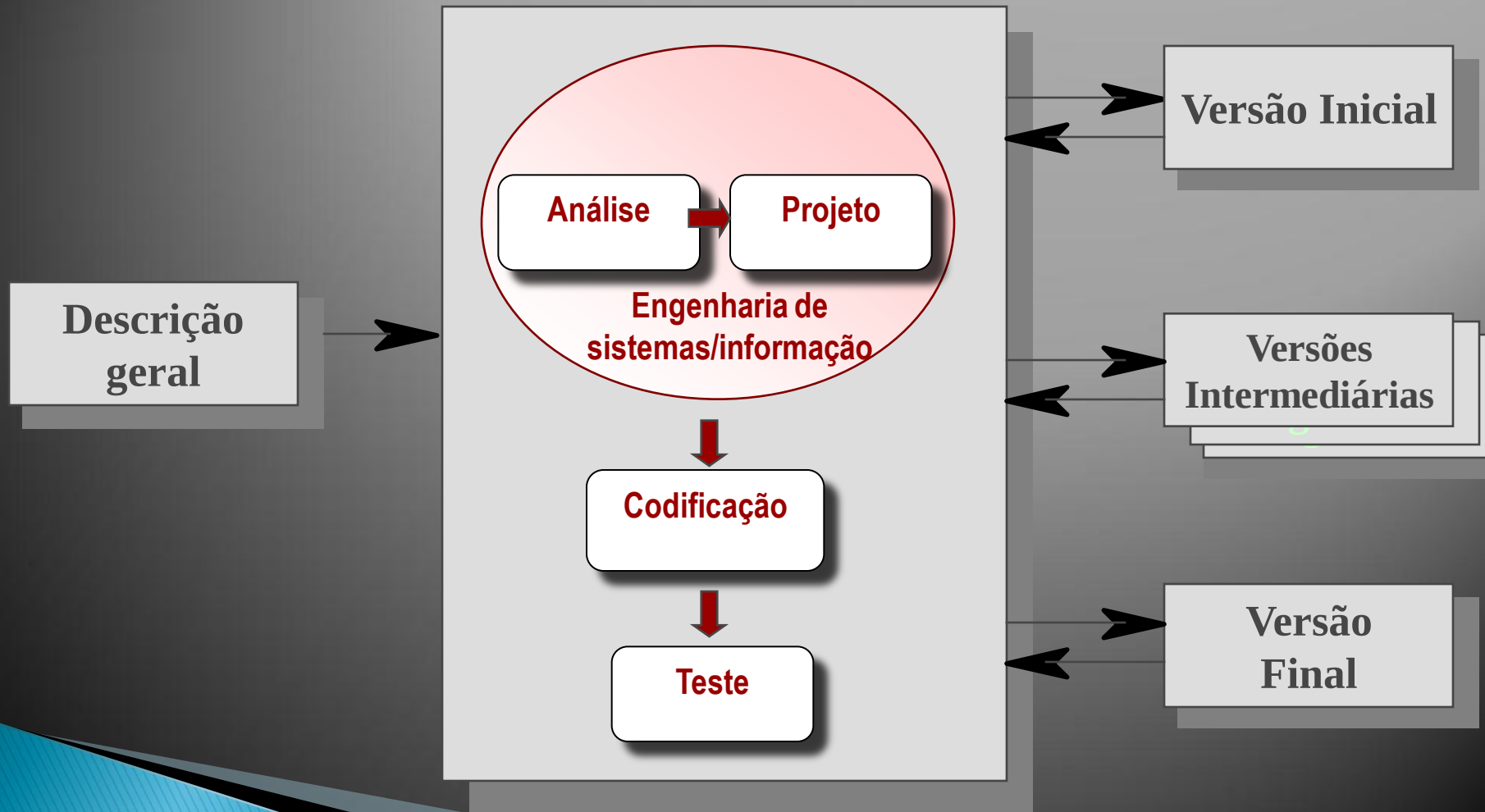
O Modelo Cascata



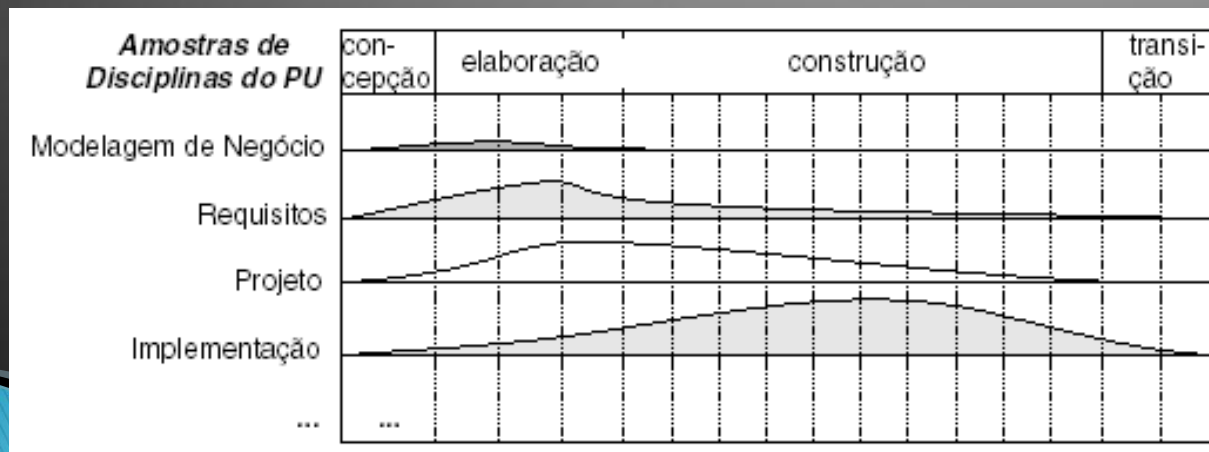
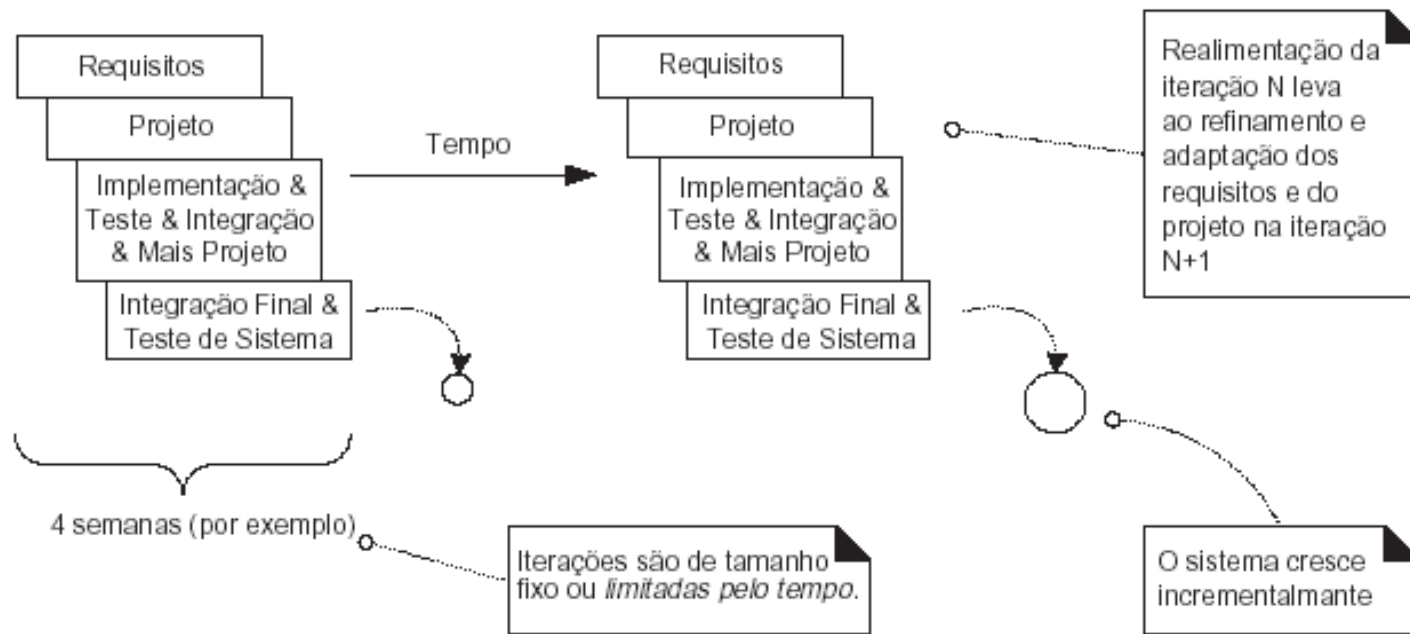
O Paradigma de Prototipação para obtenção dos requisitos



O Modelo Incremental

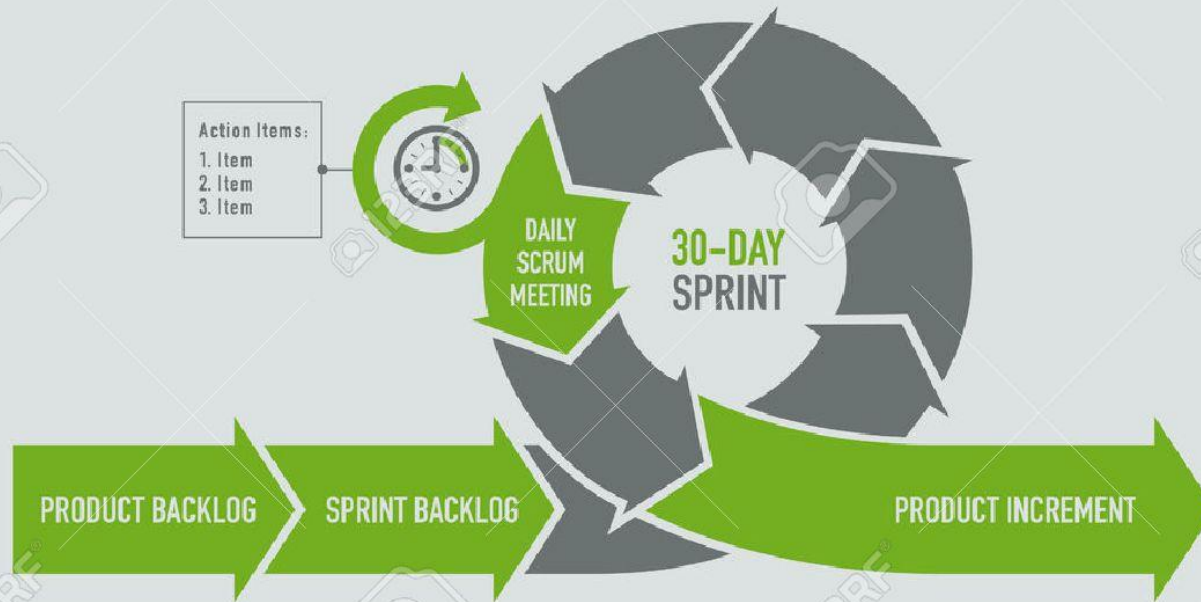


Processo Unificado



O esforço relativo nas disciplinas muda ao longo das fases. Este exemplo é uma sugestão e não deve ser tomado "ao pé da letra".

SCRUM DEVELOPMENT PROCESS



SCRUM ROLES:


Product owner

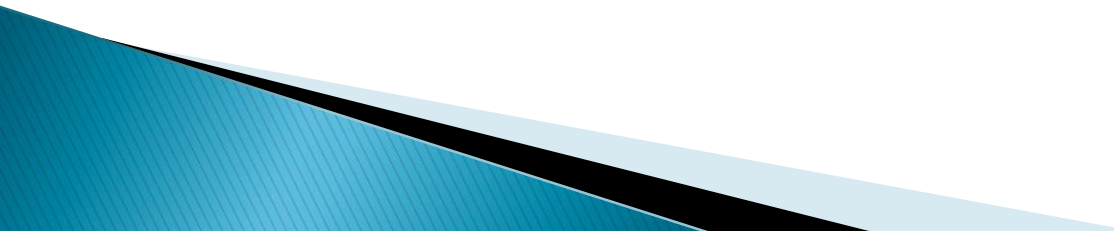

Scrum master


Team members

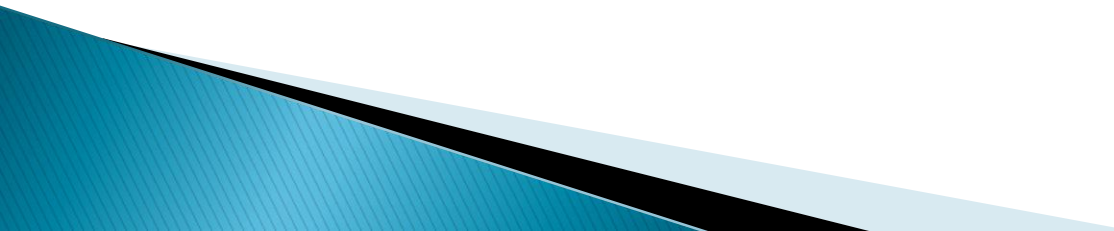

Users


Stakeholders

Reúso de Software

- ▶ O processo de desenvolvimento:
 - Tem que ser adaptado para o reúso de software
 - Ou
 - Tem que ser concebido já incluindo o reúso de software
 - ▶ Dois pontos de vista:
 - Desenvolvimento com reúso
 - Desenvolvimento para reúso
- 

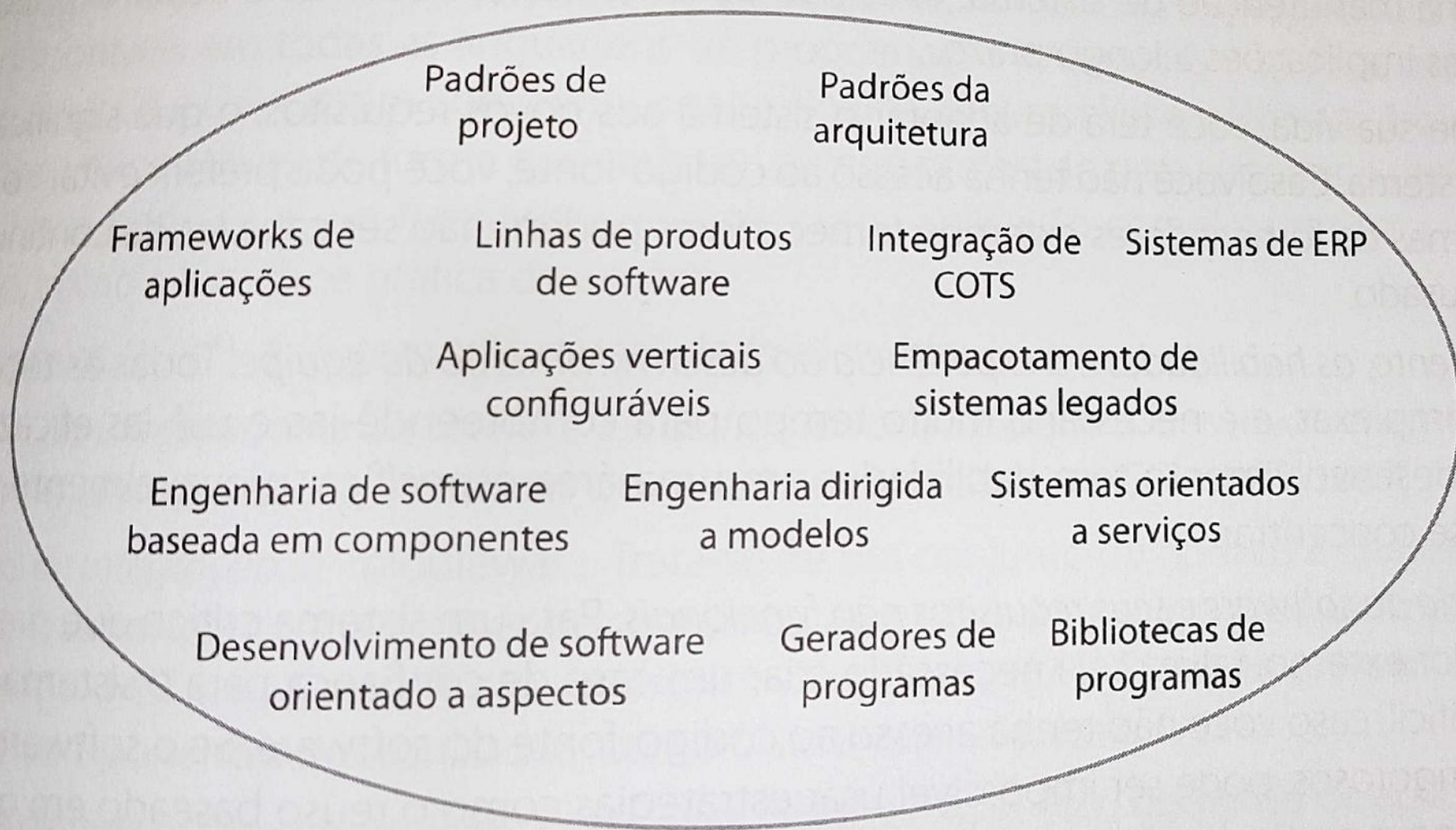
O que considerar no planejamento do reúso

- ▶ Cronograma de Desenvolvimento do Software
 - ▶ Ciclo de vida previsto para o software (Longo ou curto?)
 - ▶ Conhecimentos, habilidades e experiência da equipe de desenvolvimento.
 - ▶ Domínio da aplicação
- 

O que considerar no planejamento do reúso (Cont.)

- ▶ Importância do software e seus requisitos funcionais
 - Sistema crítico com validação externa (acesso ao código fonte?)
 - Desempenho → geradores provavelmente não atenderão.
- ▶ Plataforma sobre a qual o sistema será executado

O panorama de reúso (Sommerville)



Abordagens/ Tecnologias para Reúso

Abordagem	Descrição
Padrões de arquitetura	Padrões de arquitetura de software que oferecem suporte a tipos comuns de sistemas de aplicação são usados como base de aplicações. São descritos nos capítulos 6, 13 e 20.
Padrões de projeto	Abstrações genéricas que ocorrem em todas as aplicações são representadas como padrões de projeto, mostrando os objetos abstratos e concretos e as interações. São descritos no Capítulo 7.
Desenvolvimento baseado em componentes	Sistemas são desenvolvidos através da integração de componentes (coleções de objetos) que atendem aos padrões de modelos e componentes. São descritos no Capítulo 17.
Framework de aplicações	Coleções de classes abstratas e concretas são adaptadas e estendidas para criar sistemas de aplicação.
Empacotamento de sistemas legados	Sistemas legados (veja o Capítulo 9) são 'empacotados' pela definição de um conjunto de interfaces e acesso a esses sistemas legados por meio dessas interfaces.
Sistemas orientados a serviços	Sistemas são desenvolvidos pela ligação de serviços compartilhados, que podem ser fornecidos externamente. São descritos no Capítulo 19.
Linhas de produtos de software	Um tipo de aplicação é generalizado em torno de uma arquitetura comum para que esta possa ser adaptada para diferentes clientes.
Reúso de produto COTS	Sistemas são desenvolvidos pela configuração e integração de sistemas de aplicação existentes.

Abordagens/ Tecnologias para Reúso

Sistemas de ERP	Sistemas de grande porte que sintetizam a funcionalidade e as regras de negócios genéricos são configurados para uma organização.
Aplicações verticais configuráveis	Sistemas genéricos são projetados para poder ser configurados para as necessidades dos clientes de sistemas específicos.
Bibliotecas de programas	Bibliotecas de classe e funções que implementam abstrações comumente usadas são disponibilizadas para reúso.
Engenharia dirigida a modelos	O software é representado como modelos de domínio e modelos de implementação independentes. O código é gerado a partir desses modelos. São descritos no Capítulo 5.
Geradores de programas	Um sistema gerador incorpora o conhecimento de um tipo de aplicação, e é usado para gerar sistemas nesse domínio a partir de um modelo de sistema fornecido pelo usuário.
Desenvolvimento de software orientado a aspectos	Quando o programa é compilado, os componentes compartilhados são integrados em uma aplicação em diferentes locais. São descritos no Capítulo 21.