

Modelos de Processo de Software - continuação



Engenharia de Software
Profa. Dra. Rosana T. Vaccare Braga
1º semestre de 2017

(material produzido e atualizado pelos professores do grupo de pesquisa em Engenharia de Software do ICMC-USP)

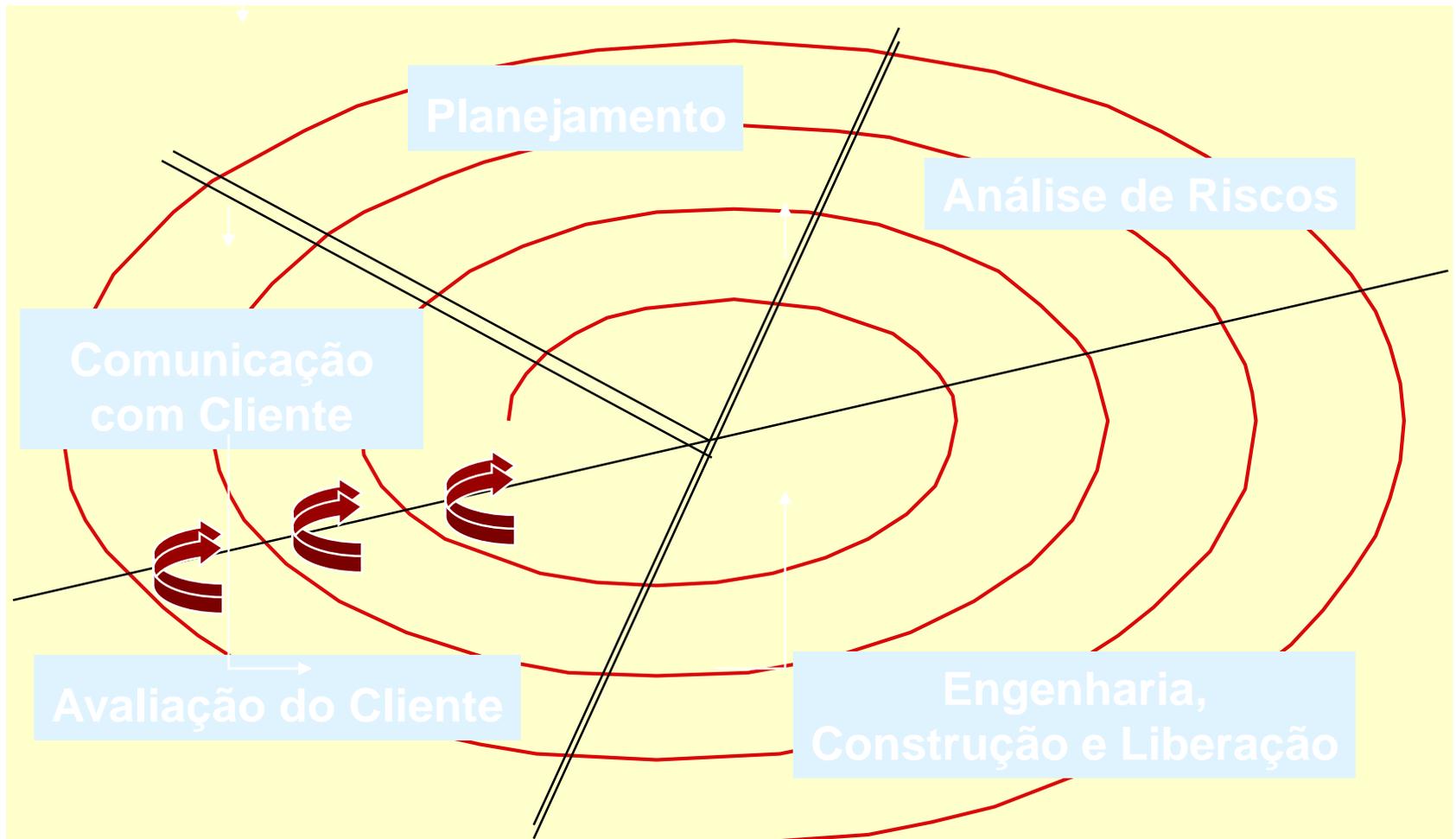
Modelos de Processo de Software

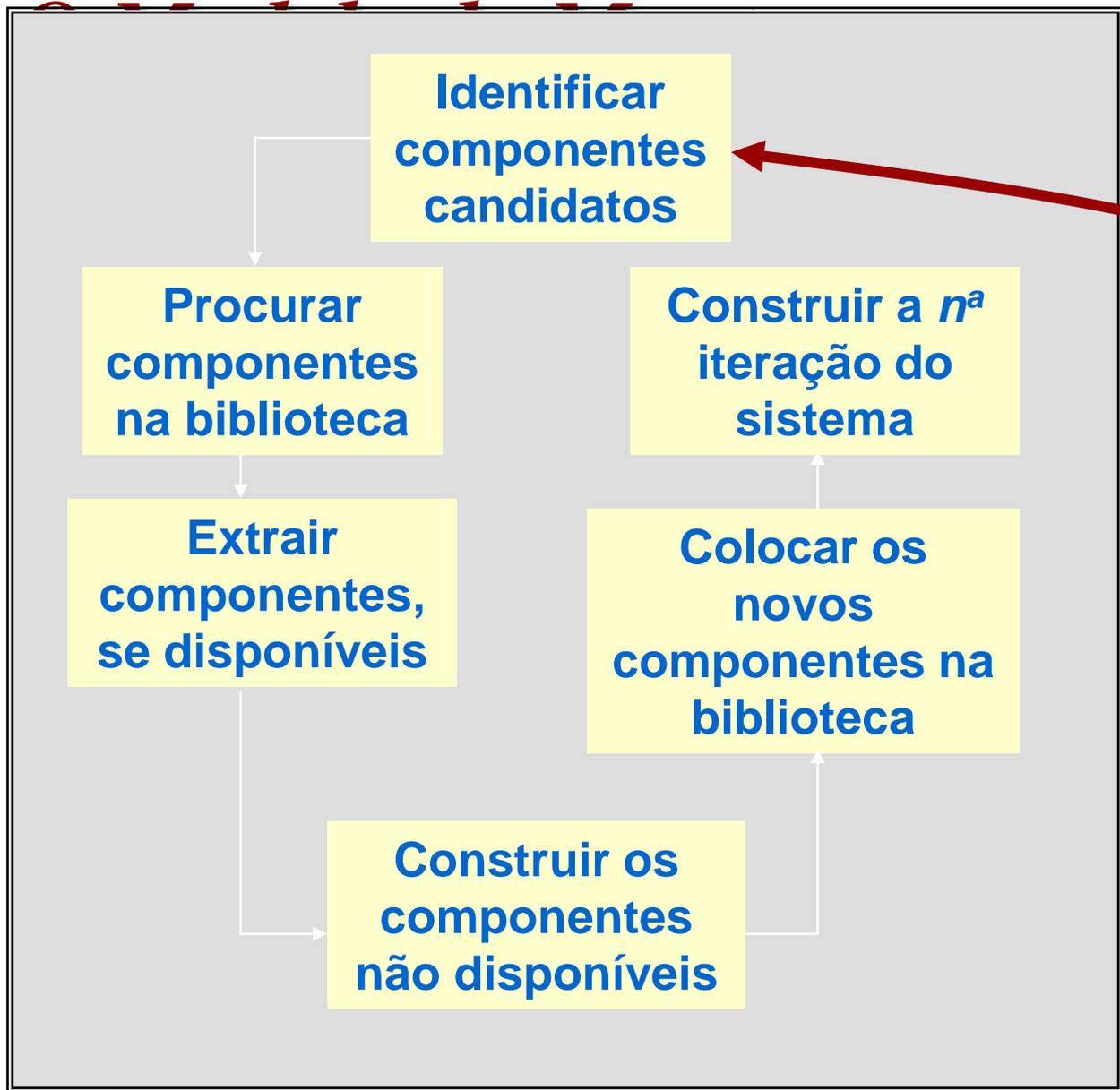
- *O Modelo Sequencial Linear*
 - *também chamado Modelo Cascata*
- *O Modelo de Prototipação*
- *O Modelo RAD (Rapid Application Development)*
- **Modelos Evolutivos de Processo de Software**
 - *O Modelo Incremental*
 - *O Modelo Ágil*
 - *O Processo Unificado – visto em APOO*
 - *O Modelo Espiral*
 - ***O Modelo de Montagem de Componentes***
 - *O Modelo de Desenvolvimento Concorrente*
- *Modelo de Métodos Formais*
- *Técnicas de Quarta Geração*

O Modelo de Montagem de Componentes

- Incorpora características de **tecnologias orientadas a objeto** no **Modelo Espiral**.
- É de natureza **evolutiva** demandando uma abordagem **iterativa** para a criação do software.
- O modelo **compõe** aplicações a partir de componentes de software “**empacotados**”.
 - Quando projetadas e implementadas apropriadamente, as classes são **reutilizáveis** em diferentes aplicações e arquiteturas de sistema.

O Modelo de Montagem de Componentes





This area features a yellow background with two light blue rectangular boxes containing white text. The top box is labeled 'de Riscos' and the bottom box is labeled 'aria, Liberação'. Several red curved arrows originate from the right side of the image and point towards the 'Identificar componentes candidatos' box in the main diagram.

O Modelo de Montagem de Componentes

- O modelo de montagem de componentes conduz ao **reúso** do software.
- A reusabilidade pode fornecer uma série de benefícios:
 - Redução no tempo de desenvolvimento.
 - Redução no custo do projeto.
 - Aumento no índice de produtividade.
 - Tais benefícios dependem da **robustez da biblioteca** de componentes.

Modelos de Processo de Software

- *O Modelo Sequencial Linear*
 - *também chamado Modelo Cascata*
- *O Modelo de Prototipação*
- *O Modelo RAD (Rapid Application Development)*
- **Modelos Evolutivos de Processo de Software**
 - *O Modelo Incremental*
 - *O Modelo Ágil*
 - *O Processo Unificado – visto em APOO*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
 - *O Modelo de Desenvolvimento Concorrente*
- ***Modelo de Métodos Formais***
- ***Técnicas de Quarta Geração***

O Modelo de Métodos Formais

- Especificar, desenvolver e verificar um sistema pela aplicação de uma **rigorosa notação matemática**.
 - Abrange um conjunto de atividades que levam à **especificação matemática formal** do software.
 - Tem como base a **transformação matemática formal** de uma especificação de sistema em um programa executável.

O Modelo de Métodos Formais

LearningOrg

FormalOrg

learningteams : \mathbb{P} *LearningTeam*

sharedvisions : \mathbb{P} *SharedVision*

sharedvisions \neq

$\forall lt : \text{LearningTeam} \mid lt \in \text{learningteams} \bullet$

$(\exists_1 te : \text{Team} \mid te \in \text{teams} \bullet$

$te = (\lambda \text{LearningTeam} \bullet \theta \text{Team}) lt)$

$\#\text{teams} = \#\text{learningteams}$

O Modelo de Métodos Formais

- No projeto...
 - Base para a **verificação do programa**, permitindo que se descubram erros que poderiam passar despercebidos.
- No desenvolvimento...
 - **Ambigüidade**, **não completitude** e **inconsistência** podem ser mais facilmente descobertas pela análise matemática.

O Modelo de Métodos Formais

- Abordagem particularmente adequada ao desenvolvimento de sistemas com rigorosas exigências de **segurança, confiabilidade e garantia**.
 - Eletrônica de aeronaves.
 - Controle de tráfego aéreo.
 - Dispositivos médicos.
 - Telecomunicações.

Modelo de Métodos Formais: Problemas

- Preocupações quanto à aplicabilidade em ambientes comerciais:
 - O desenvolvimento é **lento** e **dispendioso**.
 - Requer **treinamento extensivo**.
 - Difícil utilizar os modelos como um mecanismo de **comunicação**.
 - Clientes despreparados tecnicamente.

Cleanroom (Sala Limpa)

- **Exemplo** mais conhecido do processo de desenvolvimento formal.
 - Desenvolvimento **incremental** do software.
 - Cada estágio é desenvolvido e sua correção é demonstrada utilizando-se uma **abordagem formal**.
 - O teste de sistema visa a medir a **confiabilidade** do sistema.
 - Conduzido como um experimento estatístico.

Modelos de Processo de Software

- *O Modelo Sequencial Linear*
 - *também chamado Modelo Cascata*
- *O Modelo de Prototipação*
- *O Modelo RAD (Rapid Application Development)*
- **Modelos Evolutivos de Processo de Software**
 - *O Modelo Incremental*
 - *O Modelo Ágil*
 - *O Processo Unificado – visto em APOO*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
 - *O Modelo de Desenvolvimento Concorrente*
- *Modelo de Métodos Formais*
- ***Técnicas de Quarta Geração***

Técnicas de Quarta Geração

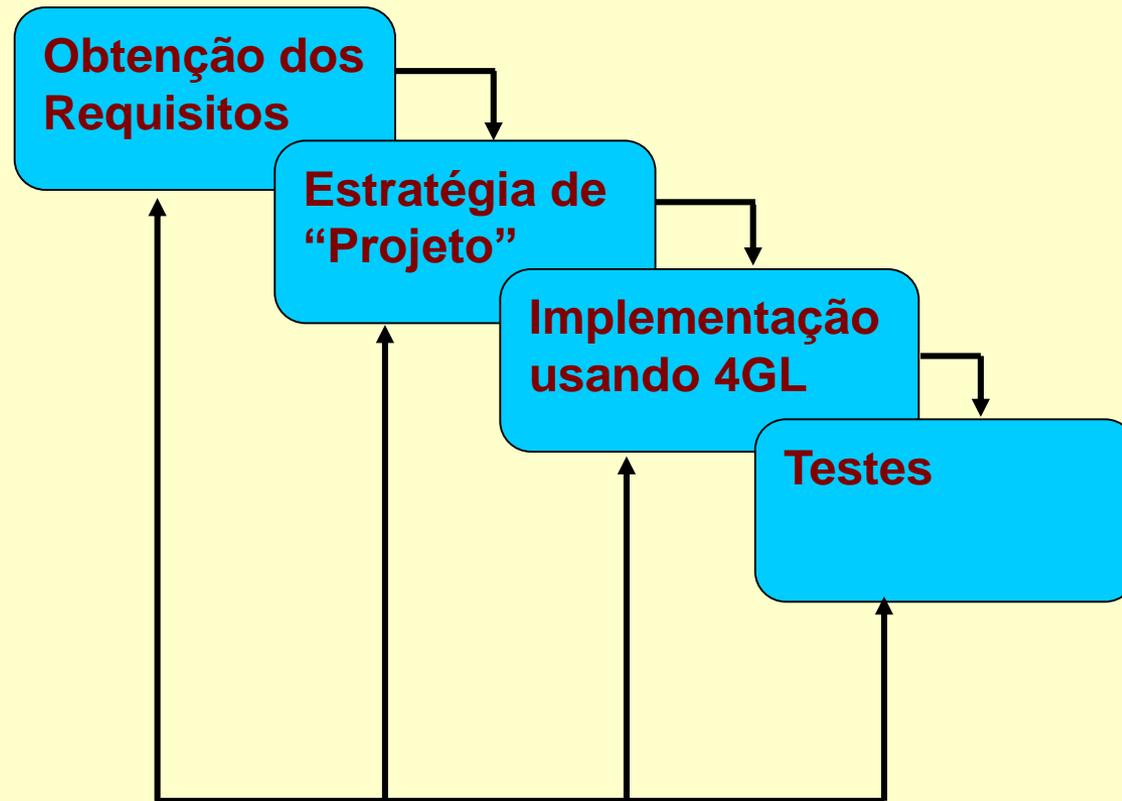
- Engloba um conjunto de ferramentas de software possibilitando que:
 - O sistema seja especificado em uma **linguagem de alto nível**.
 - O código-fonte seja gerado **automaticamente** a partir dessas especificações.

Técnicas de Quarta Geração

➤ Ferramentas de Apoio

- Linguagens não-procedimentais para:
 - Consulta de banco de dados (*SQL*).
 - Geração de relatórios (*PostScript*).
 - Manipulação de dados (*XML*).
 - Interação e definição de telas (*Delphi, Visual Basic*).
 - Geração de código.
- Capacidade gráfica de alto nível.
- Capacidade de disposição em planilhas.
- Geração automática de HTML e linguagens similares para criação de páginas Web.

Técnicas de Quarta Geração



Técnicas de Quarta Geração

Obtenção dos Requisitos

- O cliente descreve os requisitos os quais são traduzidos para um protótipo operacional.
 - O cliente pode estar inseguro quanto aos requisitos.
 - O cliente pode ser incapaz de especificar as informações de um modo que uma ferramenta 4GL possa ser utilizada.
 - As 4GLs atuais não são suficientemente sofisticadas para acomodar a verdadeira linguagem natural.

Técnicas de Quarta Geração

Estratégia de “Projeto”

- Pequenas Aplicações
 - É possível mover-se do passo de Obtenção dos Requisitos para o passo de Implementação usando uma linguagem de quarta geração.
- Grandes Projetos
 - É necessário desenvolver uma estratégia de projeto.
 - De outro modo ocorrerão os mesmos problemas encontrados quando se usa abordagem convencional (baixa qualidade).

Técnicas de Quarta Geração

Implementação usando 4GL

- Os resultados desejados são representados de modo que haja geração automática de código.
- Deve existir uma estrutura de dados com informações relevantes e que seja acessível pela 4GL.

Técnicas de Quarta Geração

Testes

- O desenvolvedor deve efetuar testes e desenvolver uma documentação significativa.
- O software desenvolvido deve ser construído de maneira que a manutenção possa ser efetuada prontamente.

Técnicas de Quarta Geração

➤ Prós

- Redução dramática no **tempo** de desenvolvimento do software.
 - Aumento de **produtividade**.

➤ Contras

- As 4GL atuais não são mais fáceis de usar do que as linguagens de programação.
- O código-fonte produzido é ineficiente.
- A **manutenibilidade** de sistemas usando técnicas 4GT ainda é questionável.

Modelos de Processo de Software

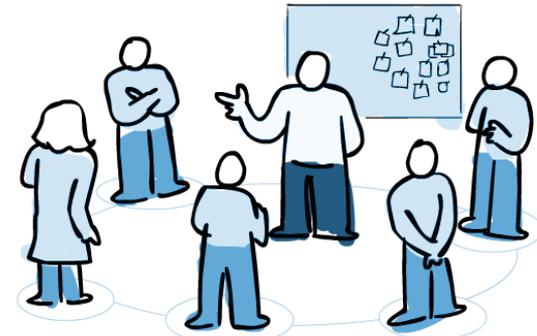
- *O Modelo Sequencial Linear*
 - *também chamado Modelo Cascata*
- *O Modelo de Prototipação*
- *O Modelo RAD (Rapid Application Development)*
- **Modelos Evolutivos de Processo de Software**
 - *O Modelo Incremental*
 - *O Modelo Ágil*
 - *O Processo Unificado – visto em APOO*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
 - *O Modelo de Desenvolvimento Concorrente*
- *Modelo de Métodos Formais*
- *Técnicas de Quarta Geração*

Modelagem Ágil

- Resultou de um esforço em vencer as fraquezas percebidas na engenharia de software convencional
 - Desenvolvedores são frágeis, diferem em habilidades, criatividade, regularidade, consistência e espontaneidade

Modelagem Ágil

- Manifesto Ágil: 2001
 - Valorizar indivíduos e interações em vez de processos e ferramentas
 - Software funcionando ao invés de documentação abrangente
 - Colaboração do cliente em vez de negociação de contratos
 - Resposta a modificações em vez de seguir um plano



Modelagem Ágil

- Equipe Ágil:
 - Auto-organização
 - Controla seu próprio destino
 - Enfatiza a comunicação e a colaboração entre todos
 - Competente
 - Foco comum
 - Capacidade de tomada de decisão
 - Habilidade de resolver problemas vagos
 - Respeito e confiança mútua

Metodologias Ágeis

- XP (Extreme Programming)
- SCRUM
- Crystal/Clear
- FDD (*Feature Driven Development*)
 - Desenvolvimento Orientado a Funcionalidades
- DSDM (*Dynamic Systems Development Method*)
 - Método Dinâmico de Desenvolvimento de Sistemas
- ASD (*Adaptive Software Development*)
 - Desenvolvimento de Software Adaptável

XP - Programação Extrema (eXtreme Programming)

➤ **Evolução** da abordagem incremental.

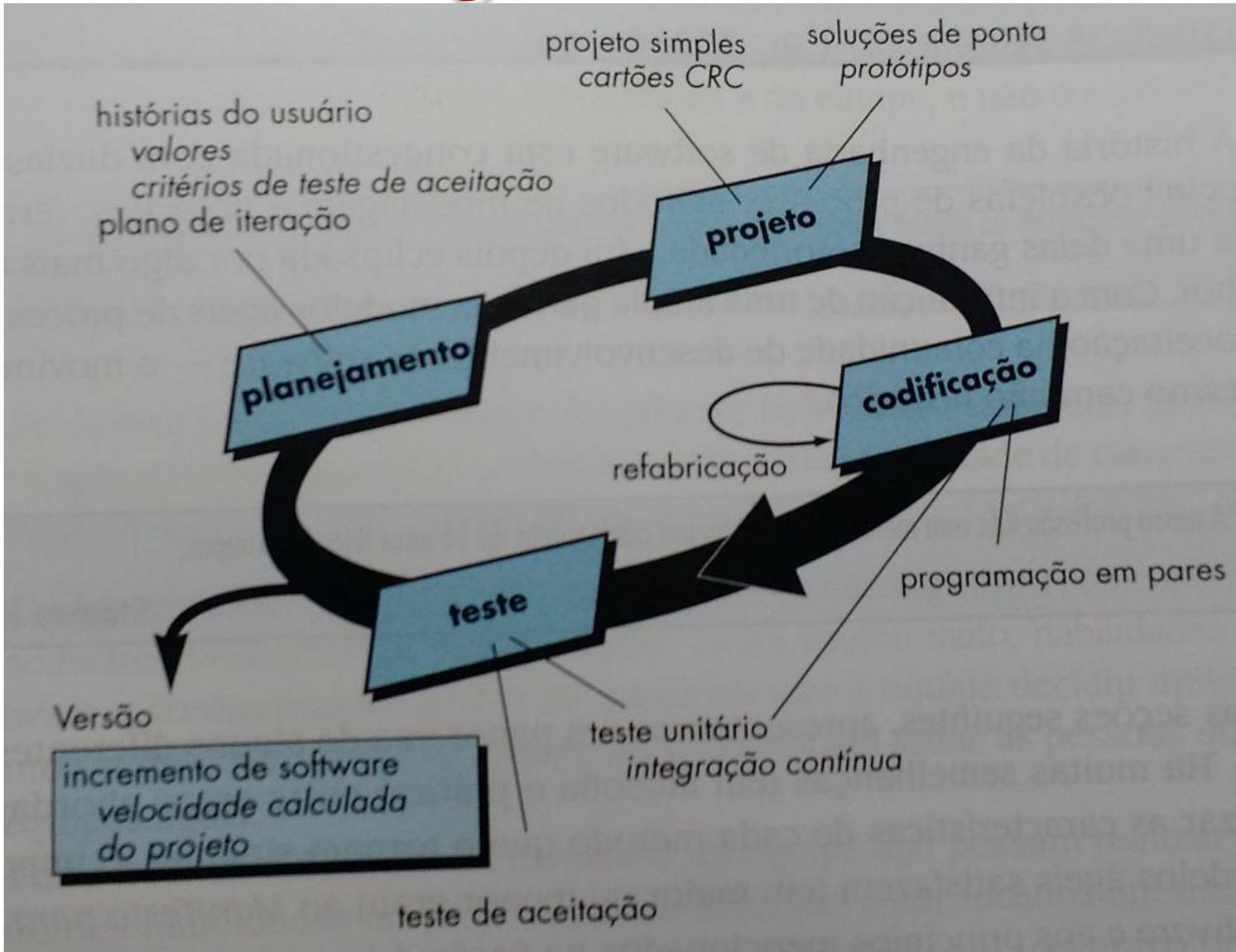
(Kent Beck, 1999)

- Voltada para equipes de até 20 pessoas, engajadas no desenvolvimento de software cujos requisitos são **vagos** ou se encontram em **constante mudança**.

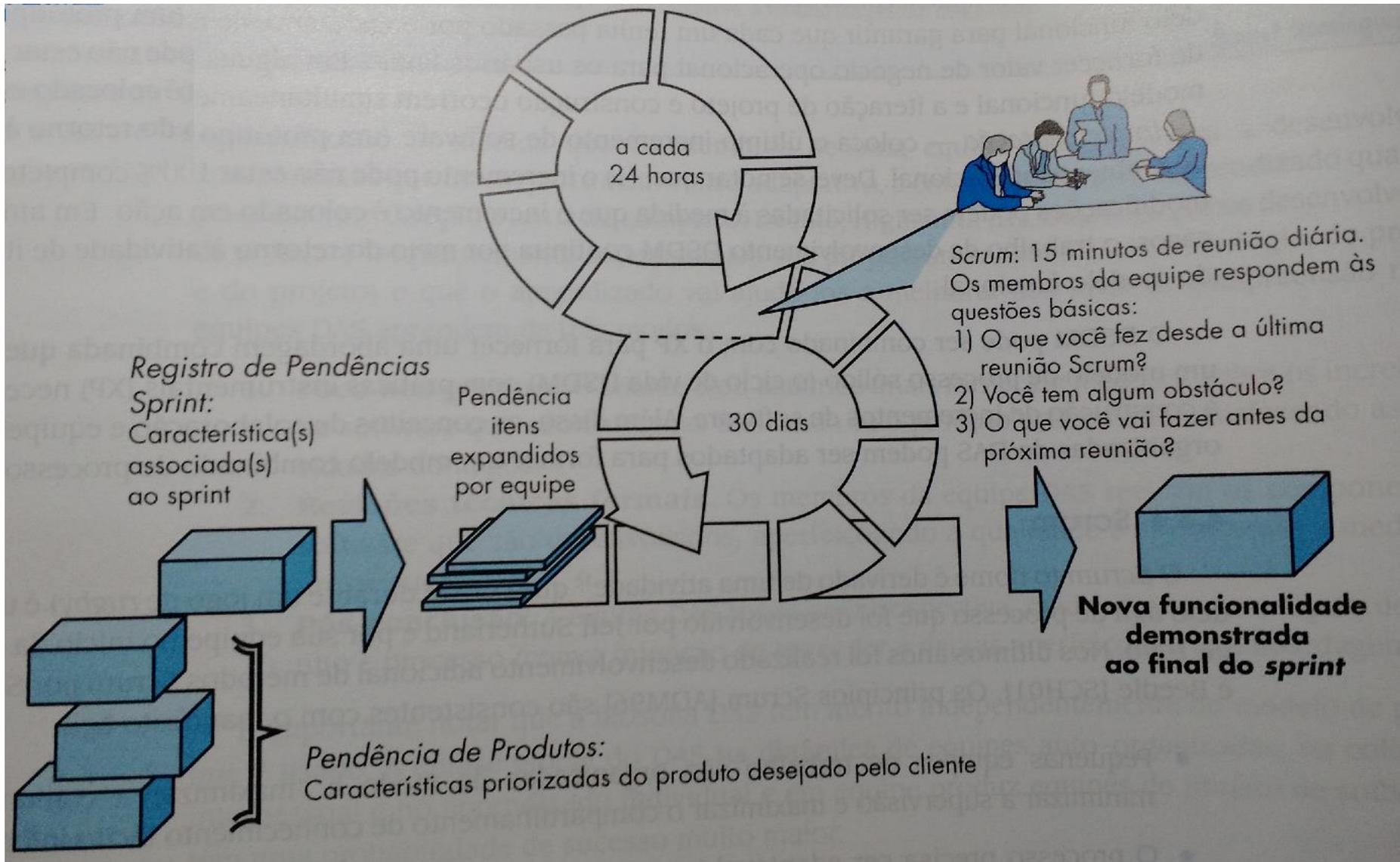
XP - Programação Extrema (eXtreme Programming)

- Desenvolvimento e entrega de **incrementos** de funcionalidade muito pequenos.
- **Envolvimento** do cliente no processo.
- Constante **melhoria** de código.
- Programação **em pares**.
- 40 horas de trabalho.
- **Refactoring**
 - Abordagem disciplinada para tornar o código de um software mais claro e de fácil manutenção, minimizando a probabilidade de inclusão de erros.
- **Test first, code later.**

Modelo Ágil XP



SCRUM



SCRUM

- Slides Bruno

Concluindo...

- *Cascata, Prototipação, Evolutivos, Formais, Ágeis, ...*
- Para escolha de um **modelo de processo de software**:
 - Natureza do projeto e da aplicação.
 - Métodos e ferramentas a serem usados.
 - Controles e produtos que precisam ser entregues.
- Não são mutuamente exclusivos e freqüentemente são usados em conjunto.

Atividade

1. Sugira o modelo de processo de software mais adequado para ser usado nos seguintes sistemas. Justifique a escolha:
 1. Um sistema para controlar a injeção automática de insulina em pacientes diabéticos
 2. Sistema de folha de pagamento para uma universidade que substitua o sistema já existente
 3. Aplicativo para smartphone para gerenciar caronas entre estudantes

