Análise de Sistemas – AULA 04
BCC Noturno - EMA908915A

Prof. Rafael Oliveira
rpaes@ic.unesp.br

Universidade Estadual Paulista
“Júlio de Mesquita Filho”
UNESP

Rio Claro 2014 (Sem 2)

Revisão Aula 03

- Praticando a Engenharia de Software

→ Como iniciar!?

- Processos de software
  - Modelos prescritivos de processo

Rev: Processo de software

unesp

- Visão Genérica:
  - Um processo é uma coleção de atividades, ações e tarefas que são realizadas quando algun produto deve ser criado.
  - Uma atividade se esforça para alcançar um amplo objetivo (ex. Comunicação com stakeholders) e é aplicado independentemente do domínio da aplicação, tamanho do projeto, complexidade de esforços, ou grau de rigor com o qual a engenharia de software é aplicada.
  - Uma ação (ex: projeto arquitetural) engloba um conjunto de tarefas que produz um grande produto (ex: um modelo de projeto arquitetural).
  - Uma tarefa foca em um pequena, mas com um objetivo bem-definido (ex: conduzindo uma unidade de teste) que produz um resultado tangível.

Rev: Processo de software

- Um framework genérico de engenharia de software engloba 5 atividades:
  - Comunicação;
  - Planejamento;
  - Modelagem;
  - Construção;
  - Implantação.

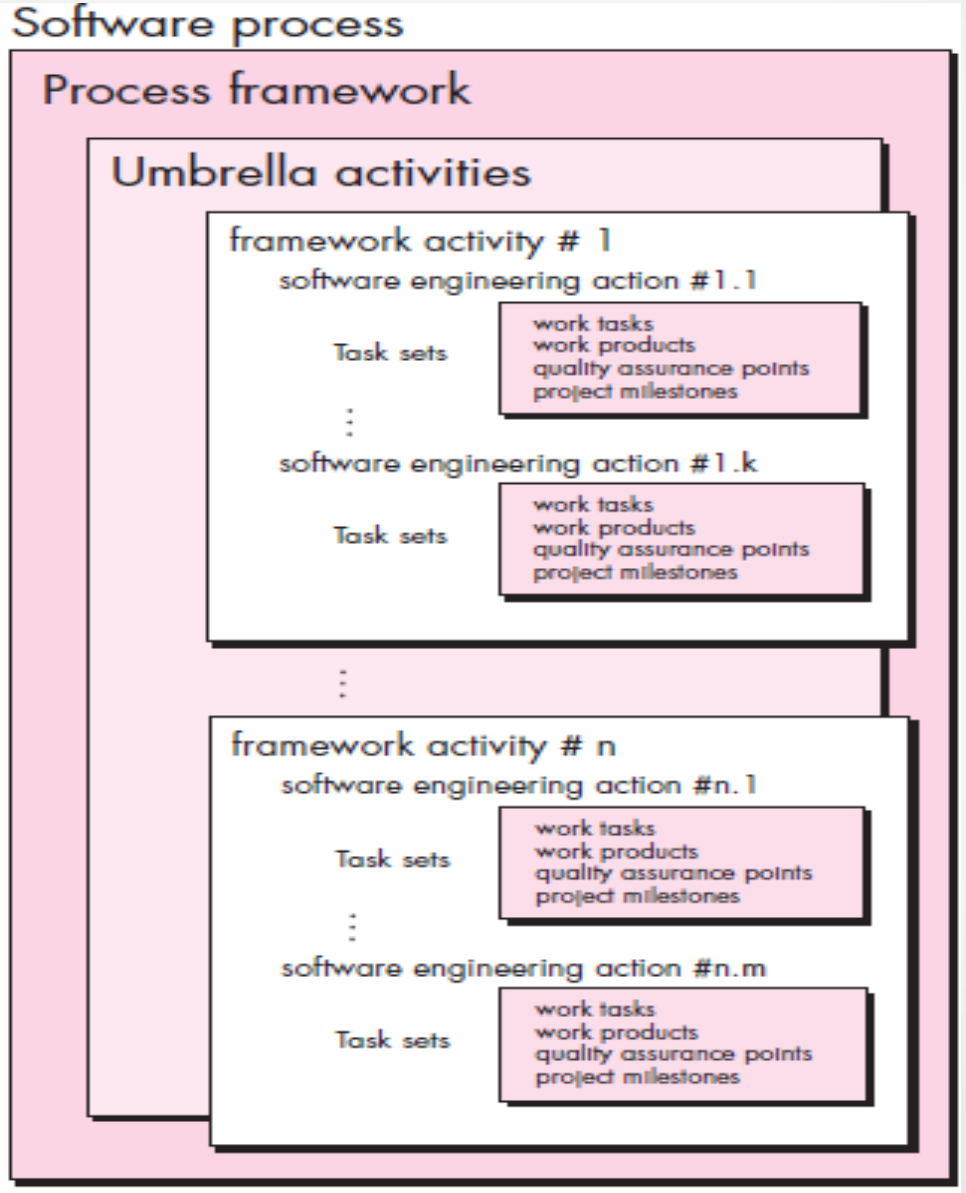
REV: Praticando Eng. Software

unesp

REVISÃO: PRATICANDO ES ...

- Partindo da essência da solução do problema!
  - (1) Entendimento do problema (comunicação e análise)
  - (2) Plano de solução (Modelagem e projeto de software)
  - (3) Execução do plano (geração de código)
  - (4) Examinação da pecisão dos resultados

REV: Processo



REV: Processo – Modelos prescritivos

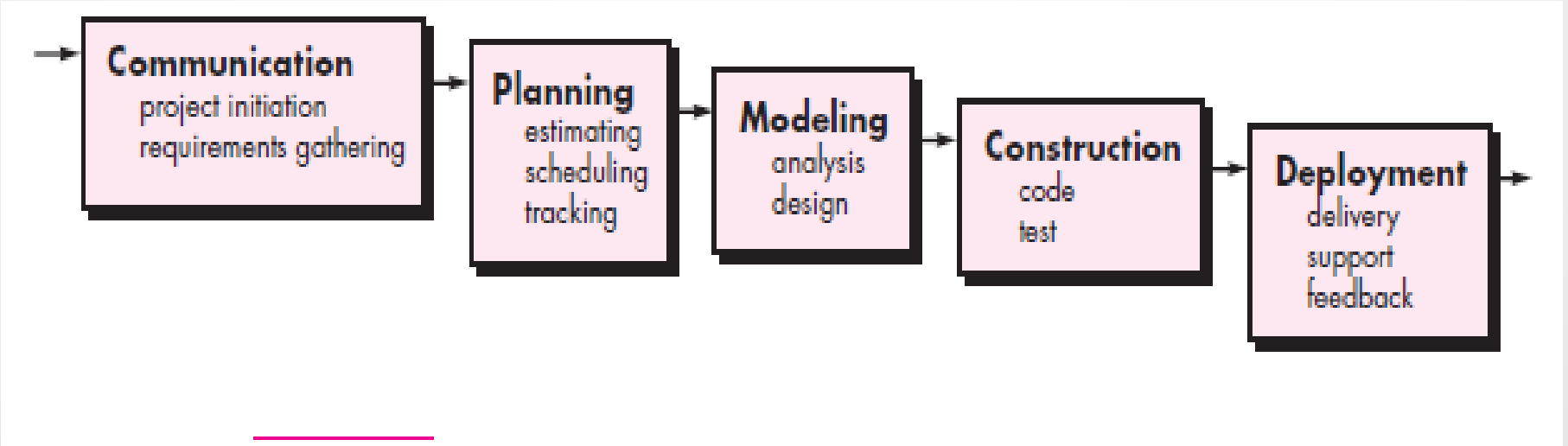
- *Modelos prescritivos:*
  - Propostos originalmente para trazer ordem ao caos que era o desenvolvimento de software;
  - A literatura indica que estes modelos trouxeram estruturas muito úteis para a Engenharia de Software

REV: Processo – Modelos prescritivos

- *Modelos prescritivos;*
  - Modelo cascata
  - Modelos de processos incrementais
  - Modelos de processos evolucionários
  - Modelos concorrentes

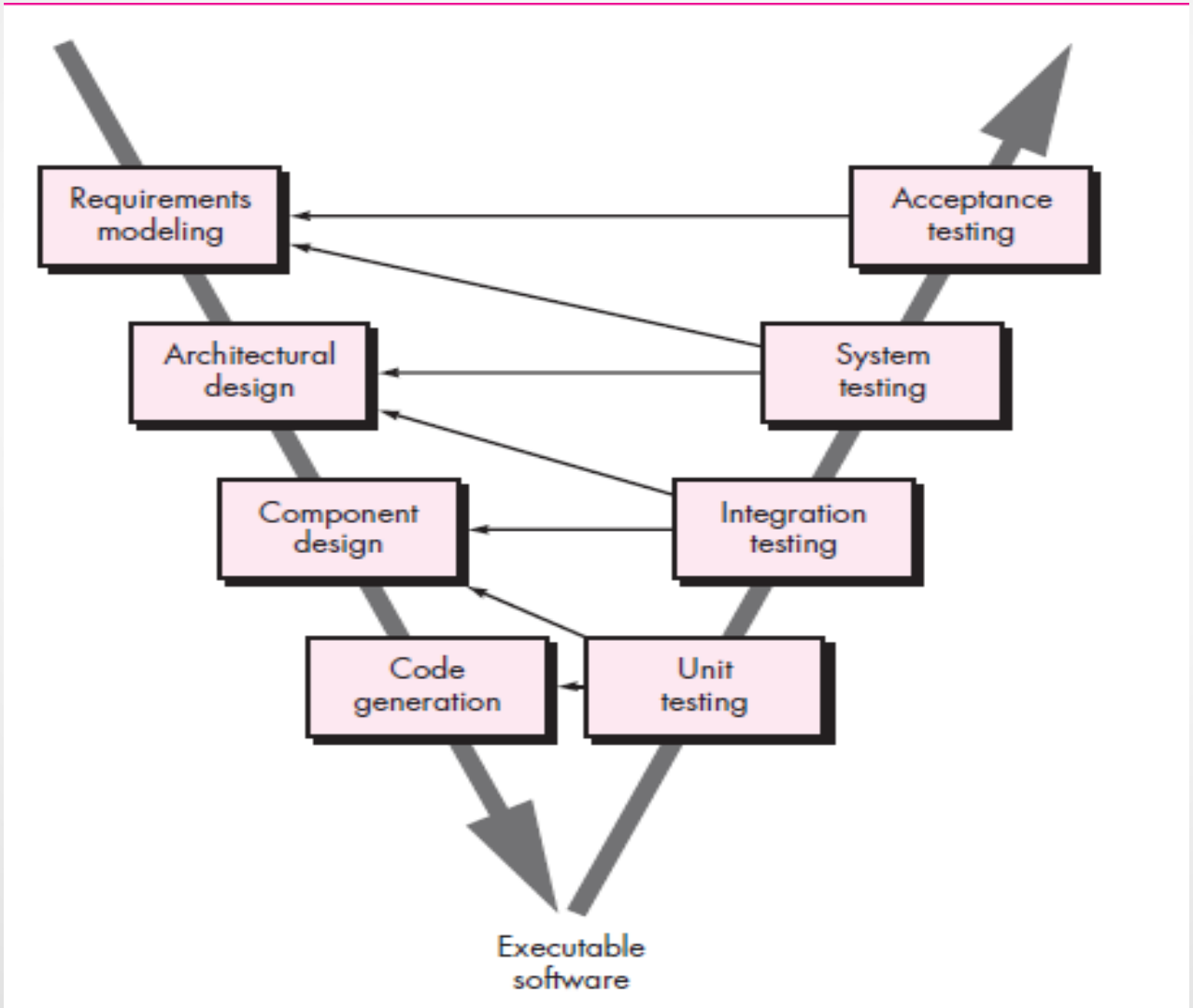
REV: Processo – Modelos prescritivos

- *Modelo cascata*



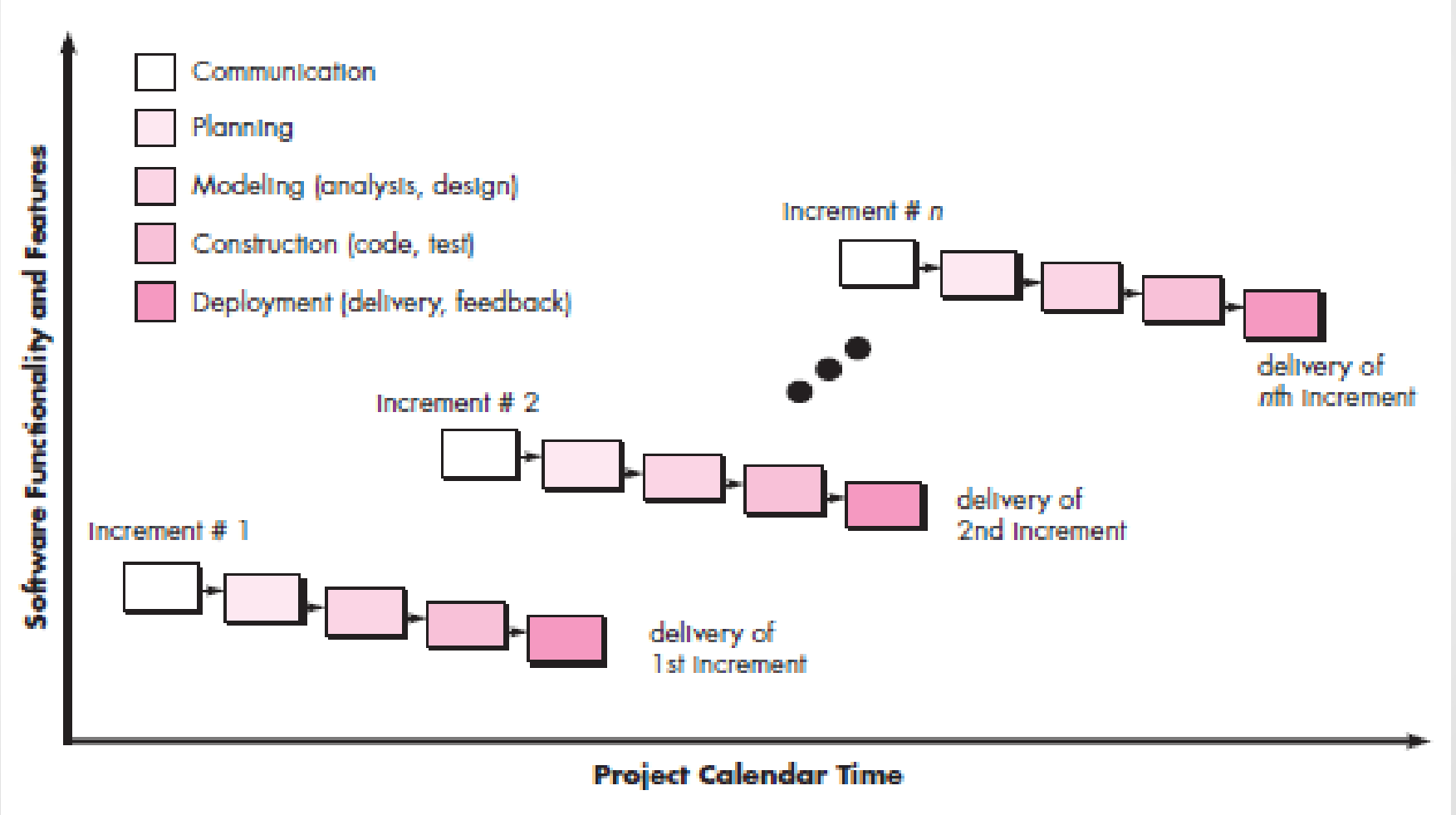
REV: Processo – Modelos prescritivos

- Modelo em V (variação do modelo cascata)



REV: Processo – Modelos prescritivos

- Modelo incremental

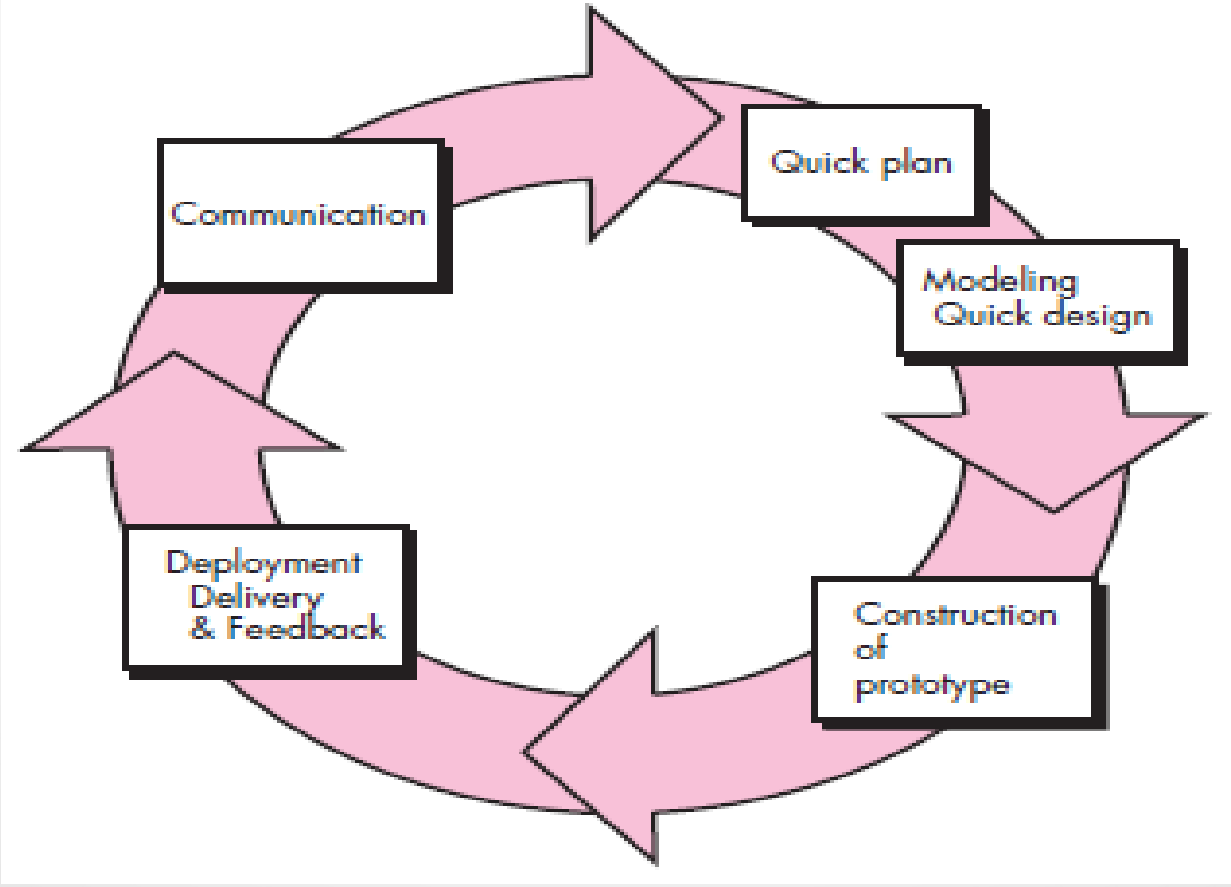


REV: Processo – Modelos prescritivos

- Processo evolucionário:
  - Exemplos:
    - Prototipagem
    - Modelo Espiral

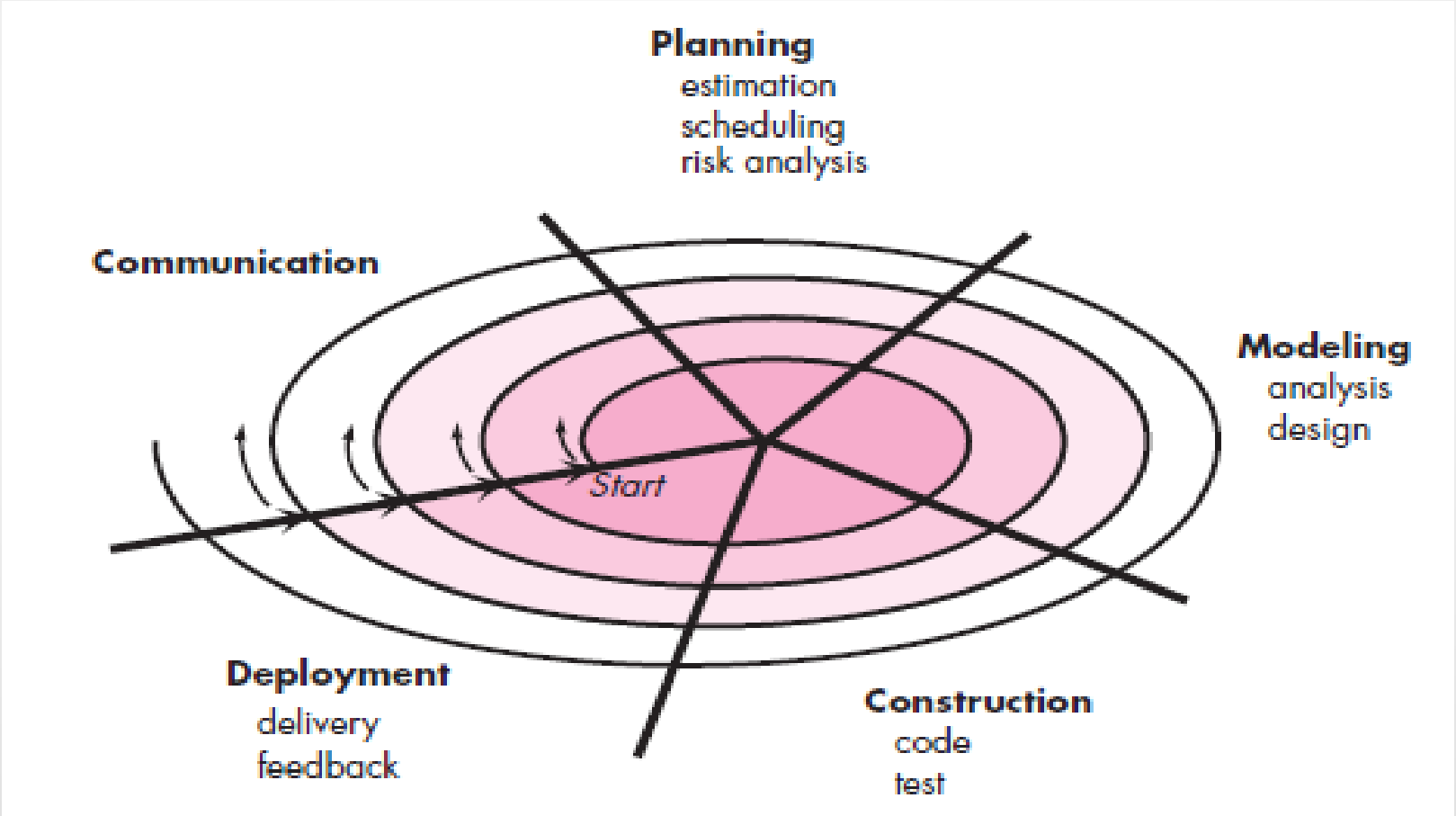
REV: Processo – Modelos prescritivos

- Processo evolucionário:
  - **Prototipagem**



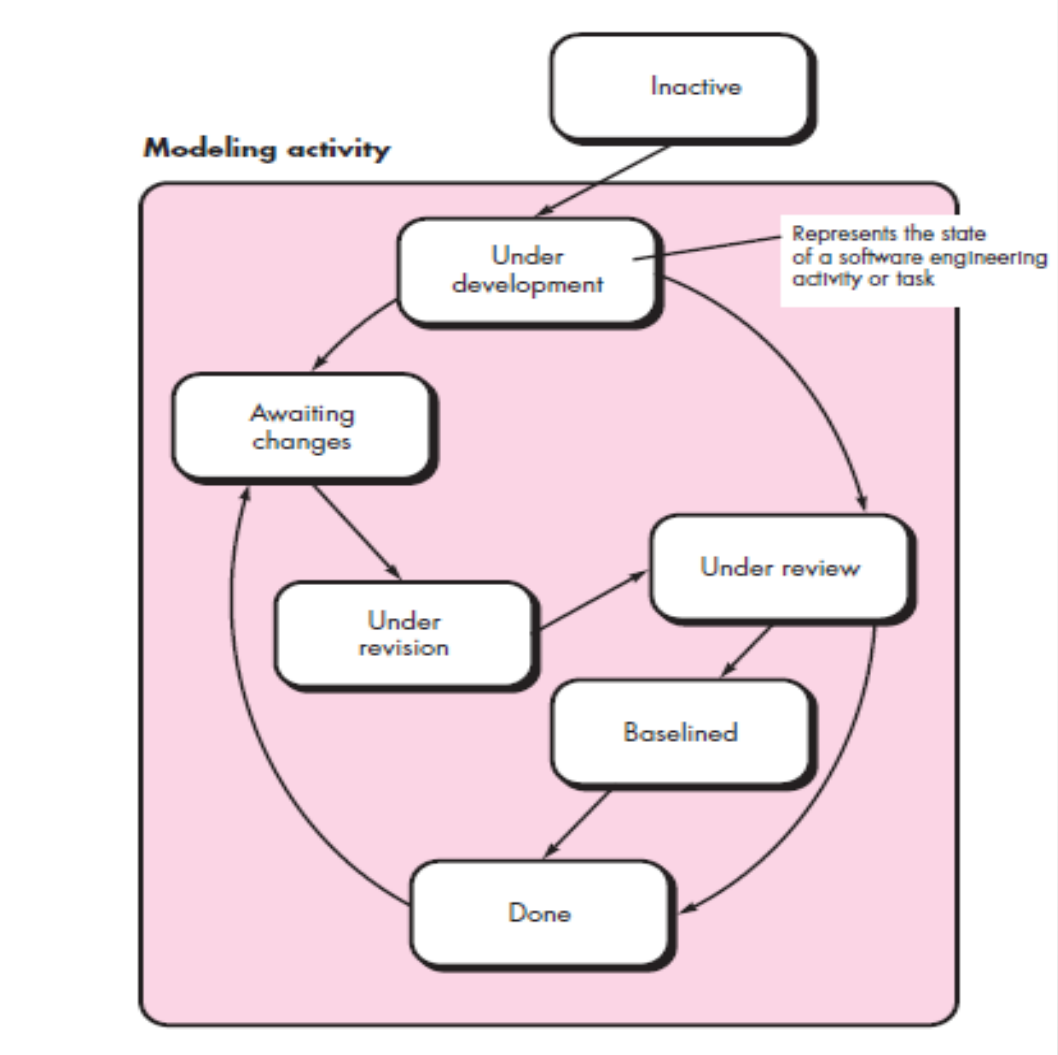
REV: Processo – Modelos prescritivos

- Processo evolucionário:
  - **Modelo espiral**



REV: Processo – Modelos prescritivos

- **Modelos concorrentes:**



AULA 4 - ROTEIRO

- Modelos ágeis
  - XP (eXtreme Programming)
  - Outros modelos
- Processos de modelos específicos (Próxima aula)
- Processo Unificado (Próxima aula)
- Eng. Requisitos (Próxima aula)

Desenvolvimento ágil

- **Desenvolvimento Ágil**

Desenvolvimento ágil

- **O que é?**
- Quem faz?
- Por que é importante?
- Quais são os passos?
- O que produz?
- Como eu garanto que foi feito corretamente?

Modelos Ágeis

- **O que é agilidade?!**
- Agilidade vem se tornando uma palavra atual do mundo de negócios associado ao software moderno
- Todos precisam ser ágeis: um bom time de desenvolvimento deve ser ágil para responder rapidamente às mudanças
- Mudanças são exatamente o que deve ser mais considerado no desenvolvimento de software nos dias atuais: mudanças no software em desenvolvimento, mudanças no time de desenvolvedores, mudanças de tecnologia
- Mudanças de todos os tipos têm impacto direto no produto de software sendo desenvolvido
- Apoio e previsão de mudanças é importante para o software desenvolvido nos dias atuais
- Mudanças estão na alma e no coração do software contemporâneo.
- Um time ágil deve considerar que o software é desenvolvido por trabalhos individuais em times e que as habilidades dessas pessoas e sua colaboração são o núcleo do sucesso do projeto de sofwtare.

Modelos Ágeis

O que é agilidade?!

- *Engenharia ágil de software* deve combinar uma filosofia e um conjunto de passos;
- A filosofia encoraja a satisfação do cliente e a precoce entrega de um software incremental;
- Filosofia: times pequenos e altamente motivados, métodos informais, mínimos produtos de E.S, simplicidade genérica de desenvolvimento;
- Os guidelines de desenvolvimento visam à entrega, ao contrário de análise e projeto (entretanto, esssa atividades não são negligenciadas);
- Por fim, é necessário uma comunicação contínua entre desenvolvedores e clientes.

Desenvolvimento ágil

- O que é?
- **Quem faz?**
- Por que é importante?
- Quais são os passos?
- O que produz?
- Como eu garanto que foi feito corretamente?

Modelos Ágeis

unesp

Quem faz?

- Engenheiros de software e outros stakeholders

  (gerentes, clientes e usuários) trabalhos juntos em um time ágil
- Time esse que é auto-organizado e controla seu próprio destino/norte/objetivos
- Um time ágil prioriza comunicação e colaboração entre todos que o servem

Desenvolvimento ágil

- O que é?
- Quem faz?
- **Por que é importante?**
- Quais são os passos?
- O que produz?
- Como eu garanto que foi feito corretamente?

Modelos Ágeis

Por que é importante?

- É importante porque o mundo dos negócios do software muda ao sabor dos ventos;
- Desenvolvimento ágil representa uma alternativa razoável para a E.S convencional em alguns projetos de sofwtare;
- Isso tem sido demonstrado pela entrega bem sucedida e rápida de sistemas.

Desenvolvimento ágil

unesp

- O que é?
- Quem faz?
- Por que é importante?
- **Quais são os passos?**
- O que produz?
- Como eu garanto que foi feito corretamente?

Modelos Ágeis

Quais são os passos?

- *Desenvolvimento poderia ser nomeado de E.S lite;*
- Um framework de passos básicos seria: communication, planning, modeling, construction, and deployment;
- Identico ao framework convencional
- Entretanto, eles devem levar a um conjunto de tarefas mínimas que pressionem o time para a entrega do software;
- Nota: Alguns podem argumentar que isso levaria ao aumento de gastos futuros com manutenção.

Desenvolvimento ágil

- O que é?
- Quem faz?
- Por que é importante?
- Quais são os passos?
- **O que produz?**
- Como eu garanto que foi feito corretamente?

Modelos Ágeis

O que produz?

- *Ambos (cliente e Engenheiros de Software) têm a mesma visão:*
  - *o único objetivo é ter uma versão funcional do software e incremental;*
  - Esse software deve ser entregue em uma data pré-estabelecida.

Desenvolvimento ágil

- O que é?
- Quem faz?
- Por que é importante?
- Quais são os passos?
- O que produz?
- Como eu garanto que foi feito corretamente?

Modelos Ágeis

Como eu garanto que foi feito corretamente?

- *Se o time ágil crê que o processo é adquado e funciona, e o time produz versões incrementais e entregáveis que satisfazem o cliente ... então o processo está sendo adequado.*

Modelos Ágeis

Agilidade vs Custo de manutenção/mudança

- Conservadores da E.S podem dizer que o desenvolvimento ágil gera custos inestimáveis;

- De um modo, geral isto pode ser considerado em partes verdadeiro:

→ obviamente, quanto mais documentação e processos detalhados de E.S mais confiável será o sofwtare

→ entretanto, atualmente os processos ágeis estão maduros e mais confiáveis (dependendo do sosftware sendo desenvolvido)

Desenvolvimento ágil

- O que é um processo ágil?
  - Do mesmo modo como é difícil fazer a elicitação de requisitos de software, é difícil predizer as prioridades dos clientes e as mudanças que elas possam acarretar nos projetos;
  - Para muitos tipos de software, projetos e implementações, são intercalados. É difícil predizer o quanto de design e modelagem será necessário.
  - Análise, projeto (design), construção e teste não são previsíveis (de um ponto de vista de planejamento) como eles deveriam ser.

Desenvolvimento ágil

- Assumidas essas premissas, surge o questionamento:
  - Como criar um processo que pode lidar com a imprevisibilidade?
  - A resposta é associada à adaptabilidade dos processos: Um processo ágil, entretanto, deve ser adaptável

Desenvolvimento ágil

- Entretanto, um processo ágil deve ser adaptável incrementalmente;
- Para alcançar uma adaptação incremental, um time ágil requer feedback do cliente (então, adaptações apropriadas podem ser feitas);
- Uma forma efetiva de ter um feedback do cliente é um protótipo operacional ou uma porção do sistema de modo operacional;
- Entretanto, uma estratégia do desenvolvimento incremental deve ser instituído;
- Incrementos de software (protótipos executáveis ou porções operacionais do sistemas) devem ser entregues em curtos períodos de tempo;
- As linhas de incremento continuam mudando durante o processo (imprevisibilidade);
- Esse processo iterativo permite que o cliente avalie o software regularmente, provendo o feedback necessário para o time de desenvolvimento, e influenciando o processo de adaptação dentro do feedback.

Desenvolvimento ágil

- *Princípios ágeis:*
  - 1 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - 2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
  - 3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - 4. Business people and developers must work together daily throughout the project.
  - 5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - 6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - 7. Working software is the primary measure of progress.
  - 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. Continuous attention to technical excellence and good design enhances agility.
  - 10. Simplicity—the art of maximizing the amount of work not done—is essential.
  - 11. The best architectures, requirements, and designs emerge from self–organizing teams.
  - 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Desenvolvimento ágil

- *Fatores humanos* (importante)*:*
  - Competência;
  - Foco comum;
  - Colaboração;
  - Habilidade de tomada de decisão;
  - Habilidade de solucionar problemas;
  - Confiança e respeito mútuo;
  - Auto-organização.

Desenvolvimento ágil

- ***XP (eXtreme Programing)***
  - Método ágil mais comumente utilizado
  - Originado no final da década de 80 por Kent Back
  - Uma versão mais atual é conhecida por IXP Industrial XP (IXP)
  - IXP refina e melhora o processo XP e processos ágeis para organizações grandes.

Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - Valores do XP
  - O processo XP
  - XP industrial
  - Discussão sobre XP

Desenvolvimento ágil

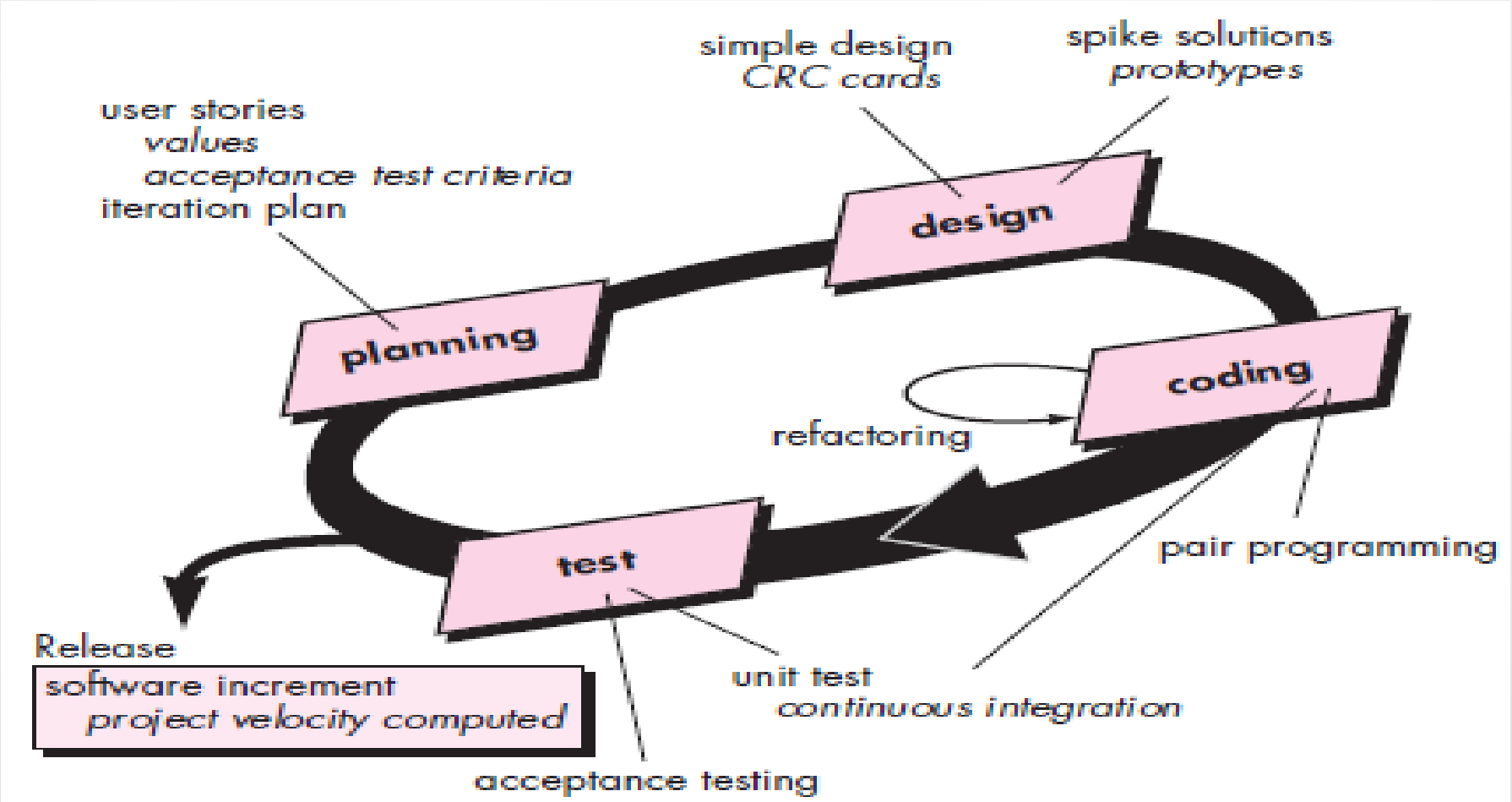
- *XP (eXtreme Programing)*
  - **Valores do XP**
  - 5 valores são essencias ao XP:
    - comunicação
    - feedback
    - coragem
    - respeito
  - Cada um desses valores são utilizados como um driver para atividades de XP, ações e tarefas.

Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - **O processo XP**
  - XP explora conceitos OO como paradigma de programação preferido;
  - XP ocorre em torno de um framework de 4 atividades: planning, design, coding, and testing
    - A Figura a seguir ilustra o processo XP. A partir dela é possível notar alguma ideia chave e tarefas associadas a cada atividade do framework.

Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - **O processo XP**



Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - **O processo XP**
  - Planejamento
  - *“ ... The planning activity (also called the planning game) begins with listening—a requirements gathering activity that enables the technical members of the XP team to understand the business context for the software and to get a broad ...”*

Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - *Design*.

    “... XP design rigorously follows the KIS (keep it simple) principle. A simple design is always preferred over a more complex representation. In addition, the design provides implementation guidance for a story as it is written—nothing less, nothing more. The design of extra functionality (because the developer assumes it will be required later) is discouraged ...”

Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - *Coding*.

    “... After stories are developed and preliminary design work is done, the team does not move to code, but rather develops a series of unit tests that will exercise each of the stories that is to be included in the current release (software increment).8 Once the unit test9 has been created, the developer is better able to focus on what must be implemented to pass the test. Nothing extraneous is added (KIS). Once the code is complete, it can be unit-tested immediately, thereby providing instantaneous feedback to the developers ...”

Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - *Testing.*

    “... I have already noted that the creation of unit tests before coding commences is a key element of the XP approach. The unit tests that are created should be implemented using a framework that enables them to be automated (hence, they can be executed easily and repeatedly). This encourages a **regression testing** strategy whenever code is modified (which is often, given the XP refactoring philosophy). As the individual unit tests are organized into a “universal testing suite” [Wel99], integration and validation testing of the system can occur on a daily basis. This provides the XP team with a continual indication of progress and also can raise warning flags early if things go awry. Wells [Wel99] states: “Fixing small problems every few hours takes less time than fixing huge problems just before the deadline ...”

Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - **XP Industrial (IXP)**
    - IXP é uma evolução organica do XP. Ele é definido como a parte minimalista do XP, centrado para o clienta e orientado a testes. IXP difere mais do geral do XP pela inclusão de gerenciamento.
    - IXP incorpora 6 novas práticas que são projetadas para ajudar a garantir que um projeto XP funciona adequadamente para projetos significativos de uma grande organização.

Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - **Discussão sobre XP**
  - **Volatilidade de requisitos**
    - Como o cliente é um membro ativo dentro do time XP, a mudança de requisitos é solicitada informalmente;
    - Como consequência, o escopo do projeto definido anteriormente pode ser alterado para acomodar necessidades atuais;
    - Proponentes argumentam que isso ocorre independentemente do processo que é aplicado e que o XP promove mecanismos de controle de escopo não formais/recomendados

Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - Discussão sobre XP
  - **Conflitos de necessidades de clientes**

    Muitos projeto tem vários clientes, com suas próprias necessidades. No XP, o time tem tarefas para assimilar e normalizar as necessidades de diferentes clientes. O que seria uma preocupação extra e fora do escopo;

Desenvolvimento ágil

- *XP (eXtreme Programing)*
  - Discussão sobre XP
  - **Requisitos são expressados informalmente**
    - Histórias de usuários e testes de aceitação são os únicos manifestos explícitos dos requisitos em XP. Críticos do método consideram que um modelo mais formal ou, então, uma especificação é muitas vezes necessária para evitar omissões, inconsitências e erros não cobertos antes do sistemas.
    - Pro-XPs consideram que as mudanças da natureza dos requisitos fazem esse modelos e especificações obsoletas assim que elas são desenvolvidas.

Desenvolvimento ágil

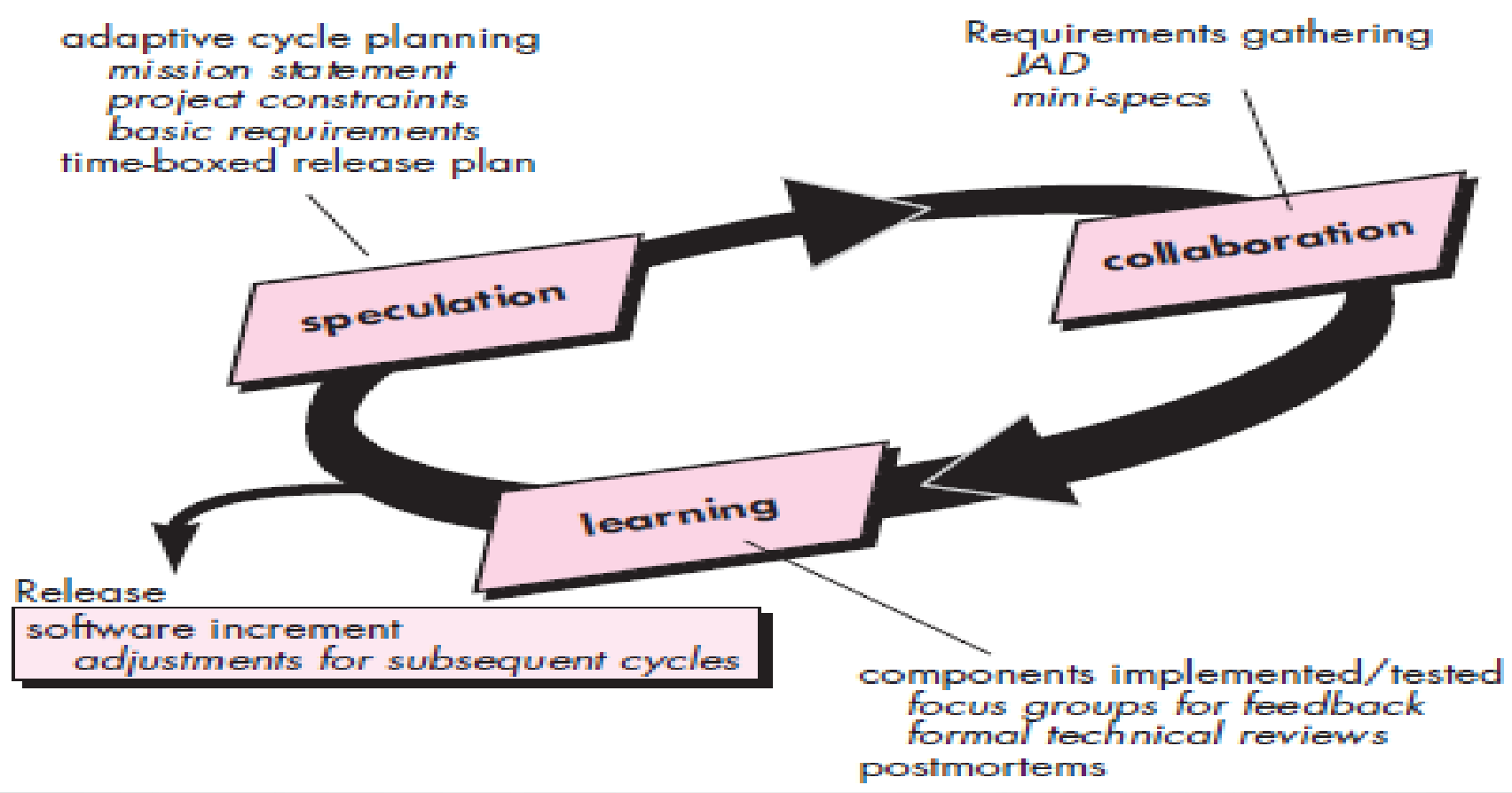
- *XP (eXtreme Programing)*
  - **Discussão sobre XP**
  - **Necessidade de projeto/modelagem formal:**
    - XP desconsidera a necessidade de projetos arquiteturais.
    - Críticos consideram que quando sistemas complexos são construídos, projetos/modelagens devem ser enfatizados para assegurar que a estrutura geral do sofwtare irá prover qualidade de manutenibilidade;
    - Pró-XP sugerem que a natureza do modelo incremental limita a complexidade e, então, reduz a necessidade de grandes modelagens.

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Adaptive Software Development (ASD)
  - Scrum
  - Dynamic Systems Development Method (DSDM)
  - Crystal
  - Feature Drive Development (FDD)
  - Lean Software Development (LSD)
  - Agile Modeling (AM)
  - Agile Unified Process (AUP)

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - **Adaptive Software Development (ASD)**



Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Scrum
    - O nome é derivado de uma atividade que ocorre durante partidas de rugby (reinício de jogos)
    - Scrum é um método de desenvolvimento ágil cunhado na década de 90.

Desenvolvimento ágil

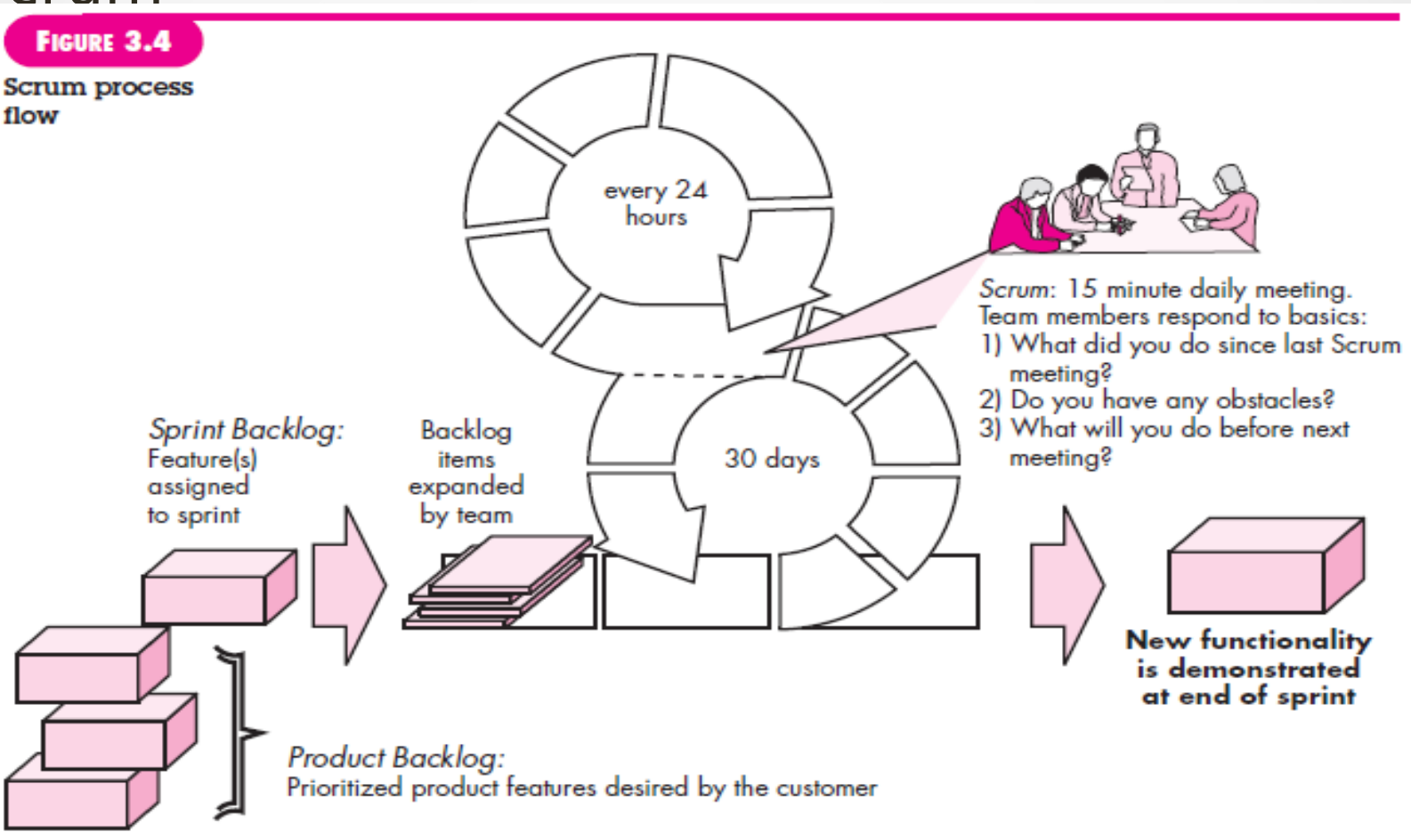
- *Outros modelos de processos ágeis*
  - Scrum
    - Princípios de scrum consistem de um processo que incorpora o seguinte frameqork de atividades: *requirements, analysis, design, evolution, and delivery.*
    - No qual cada atividade e tarefas ocorrem com um padrão de processo chamado “sprint”
    - O trabalho conduzido dentro de um sprint (o número de sprints para cada atividade do framework irá variar dependendo da complexidade e tamanho)
    - Os sprints são adaptáveis ao problema em mãos e são comumente modificados em tempo real pelo time Scrum.
    - O fluxo geral do processo é definido na figura do slide seguinte:

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Scrum

FIGURE 3.4

Scrum process flow



Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Scrum
    - Scrum enfatiza o uso de um conjunto de padrões de processos que têm de prover o projeto em uma timeline, mudando requisitos e negócios criticamente;
    - Cada um desses padrões de processo definem um conjunto de ações de desenvolvimento:
      - Backlog (acúmulo), Sprints, Mudanças e Reuniões

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Scrum
    - Backlog:
    - Uma lista prioritária de requisistos de projetos ou características que promovem o valor do négocio do cliente. Itens podem ser adicionados ao backlog a qualquer momento (desse modo, mudanças são introduzidas).
    - O produto gerencia o acesso ao backlog a atualiza as prioridades como requisitos.

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Scrum
    - Sprints:
    - Consiste das unidades de trabalho que são requeridas para atingir um requisito definido no backlog e que devem ser ajustado de um pré-definido slot de tempo (tipicamente 30 dias)
    - Mudanças (exemplo, itens do backlog) são introduzidos durante o sprint. Então, o sprint permite aos times de membros para trabalhar de forma rápida, mas em um ambiente estável.

Desenvolvimento ágil

unesp

- *Outros modelos de processos ágeis*
  - Scrum
    - Reuniões Scrum —
    - São curtas (15 minutos) realizadas diariamente pelo time scrum
    - 3 questões básicas são perguntadas para todos:
      - What did you do since the last team meeting?
      - What obstacles are you encountering?
      - What do you plan to accomplish by the next team meeting?

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Scrum
    - Reuniões Scrum:
    - Um líder do time, chamado de Scrum Master, preside a reunião para aferir a evolução de cada pessoa do time.
    - A reunião ajuda o time a cobrir potenciais problemas o quanto antes.
    - Adicionalmente, essas reuniões diárias são conhecidas como “Socialização do conhecimento” e elas promovem a auto-organização da estrutura do time de Scrum.

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Scrum
    - Demos—
    - entrega um sofwtare incremental para o cliente de modo que esse produto que foi implementado seja demostrado e avaliado pelo cliente;
    - O demo pode nao conter todas as funcionalidades planejadas inicialmente;
    - Mas ele estima que tais atividades sejam entregues em determinado slot de tempo dentro do projeto.

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - **Dynamic Systems Development Method (DSDM)**
    - É uma abordagem de desenvolvimento ágil que promove um framework de atividades de construção e manutenção de sistemas em um ambiente de desenvolvimento prototipado e controlado.
    - A filosofia adotada é que 80% da aplicação pode ser entregue em 20% do tempo que seria necessário para a entrega de 100% dessa aplicação.

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Crystal
  - É uma estratégia de modelagem ágil que considera a limitação de recursos e vê o desenvolvimento como um jogo de invenção e comunicação, com um objetivo primário de entregar software úteis e funcionais. O objetivo secundário e definir novas regras para o próximo jogo.

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Feature Drive Development (FDD)
  - Feature Driven Development (FDD) é um projeto prático para modelagem de aplicações OO. É, basicamente, um processo adaptativo, ágil que pode ser aplicados em projetos de tamanho moderado ou grandes.

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Lean Software Development (LSD)
  - Lean Software Development (LSD) adotou princípios de eliminar desperdícios, implemetar com qualidade, criar conhecimento, entrega rápida, respeito aos membros e otimizar o todo.

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Agile Modeling (AM)
  - AM (específico para sistemas complexos) defende que para o desenvolvimento de sistemas complexos e difícil compreensão, 3 regras devem ser adodatas:
    - Todos devem entender as necessidades dos envolvidos;
    - O problema pode ser particionado entre pessoas para ser solucionado;
    - A qualidade pode ser aferida conforme o sistema é implementado.

Desenvolvimento ágil

- *Outros modelos de processos ágeis*
  - Agile Unified Process (AUP)
  - AUP adota a estratégia de ser serial para sistemas complexos/grandes e iterativo para sistemas pequenos
  - As fases do framework de atividades são:
  - inception, elaboration, construction, and transition
  - AUP prevê uma estratégia para que a cobertura e evolução do projeto seja aferida a qualquer momento.

Exercícios

- 1 Descreva agilidade com suas próprias palavras.
- 2 Por que um processo iterativo faz com que as mudanças no software sejam mais bem absorvidas? Todo processo ágil é iterativo (explique)?
- 3 Faça um framework de atividades genérico para um processo ágil. Explique as possíveis tarefas em cada atividade dentro do seu framework.

Exercícios

4 Você é um analista de sofwtare de uma empresa que desenvolve sistemas comerciais, sugira mais um principio ágil que possa ajudar o seu time a ser ainda mais dinâmico.

5 Por que requisitos mudam tanto dentro de um processo de desenvolvimento de software que envolva o cliente ativamente?

6 A maioria dos processos ágeis requer comunicação pessoal. Ainda nos dias atuais, membros de um time de sofwtare e seus clientes podem estar geograficamente separados. Você acha que a separação geográfica pode ser um problema para o desenvolvimento de software utilizando algum método ágil!? Você pode sugerir algum de evitar potências problemas da distância?

Exercícios

7 Descreva com suas próprias palavras o desenvovimento ágil nos dias atuais. Elenque os modelos de processos ágeis que você conhece.

8 Exemplifique, utilizando um software comercial qualquer a ser desenvolvido, como seria um framework de atividades e tarefas para a implementação de tal software utilizando XP.