Centro Universitário FMU

Disciplina:

Engenharia de Software I
Aula 5
Bacharelado em Ciência da Computação
Tecnologia em Análise e Desenvolvimento de Sistemas

Prof.: Celso Eduardo Guimarães
celso.guimaraes@fmu.br

FMU
CENTRO UNIVERSITÁRIO

Processo Unificado Histórico

- Anos 1990
- James Rumbaugh
- Grady Booch
- Ivar Jacobson
  - **Trabalharam em um Processo Unificado como modelo para desenvolvimento de software.**
  - **A proposta foi de integrar as melhores características de cada um de seus métodos individuais de análise e projeto orientados a objetos.**
  - **Adotaram características adicionais propostas por outros especialistas em modelagem orientada a objetos.**

Processo Unificado Histórico

❑UML

- O trabalhou teve como fruto a UML - uma ***linguagem de modelagem unificada*** com uma notação robusta para a modelagem e o desenvolvimento de sistemas orientados a objetos.

❑1997

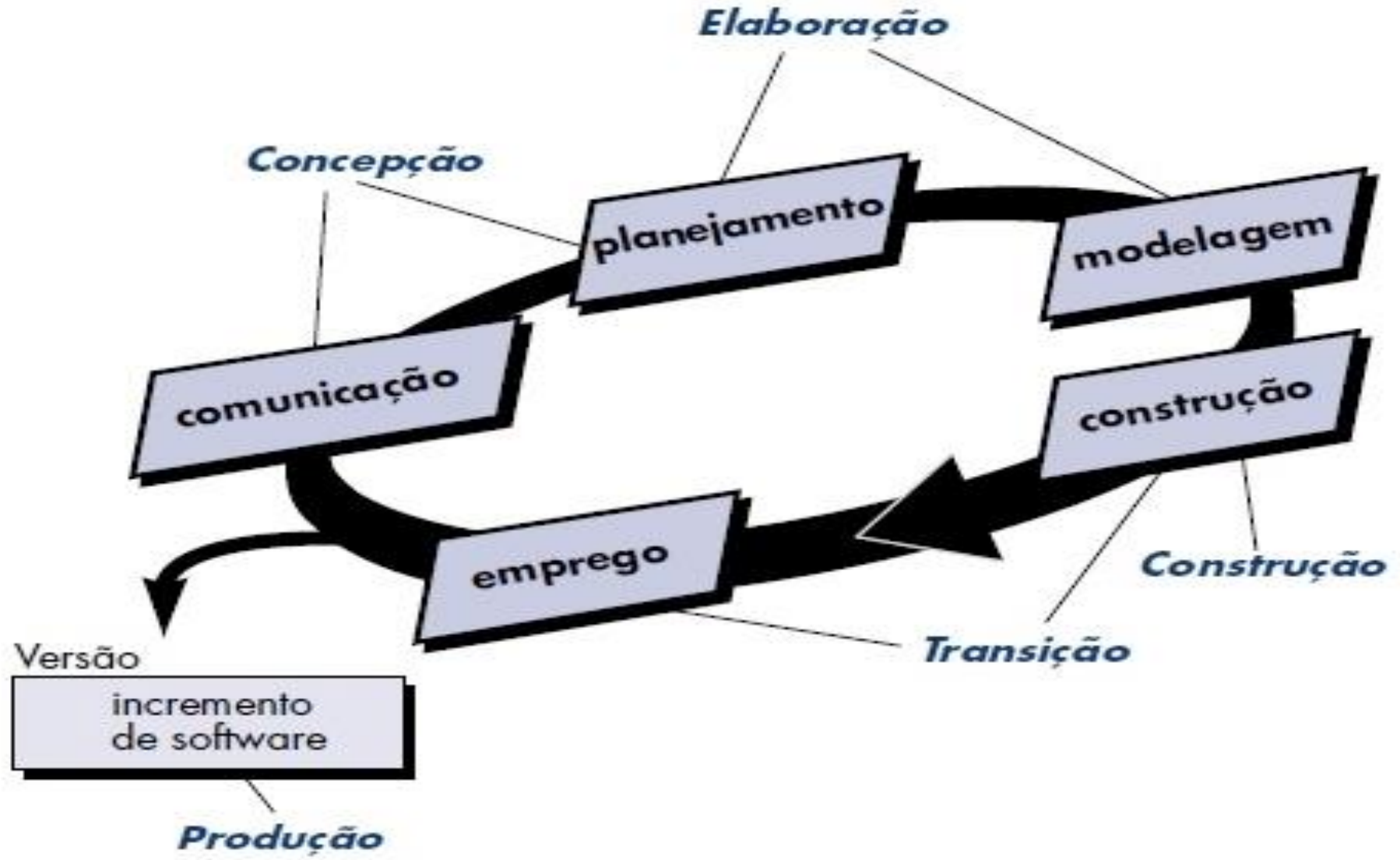
- **UML tornou-se um padrão da indústria em desenvolvimento de software orientado a objetos.**

FMU
CENTRO UNIVERSITÁRIO

Processo Unificado Histórico

- **RUP**
  - **Abreviação de Rational Unified Process , tornou-se um processo proprietário da Rational Software Corporation.**
- **2003**
  - **IBM compra a Rational e o processo passar a ser chamado de IRUP.**
  - **Todo o framework IRUP é mantido e fornecido pela IBM.**

Fases do Processo Unificado



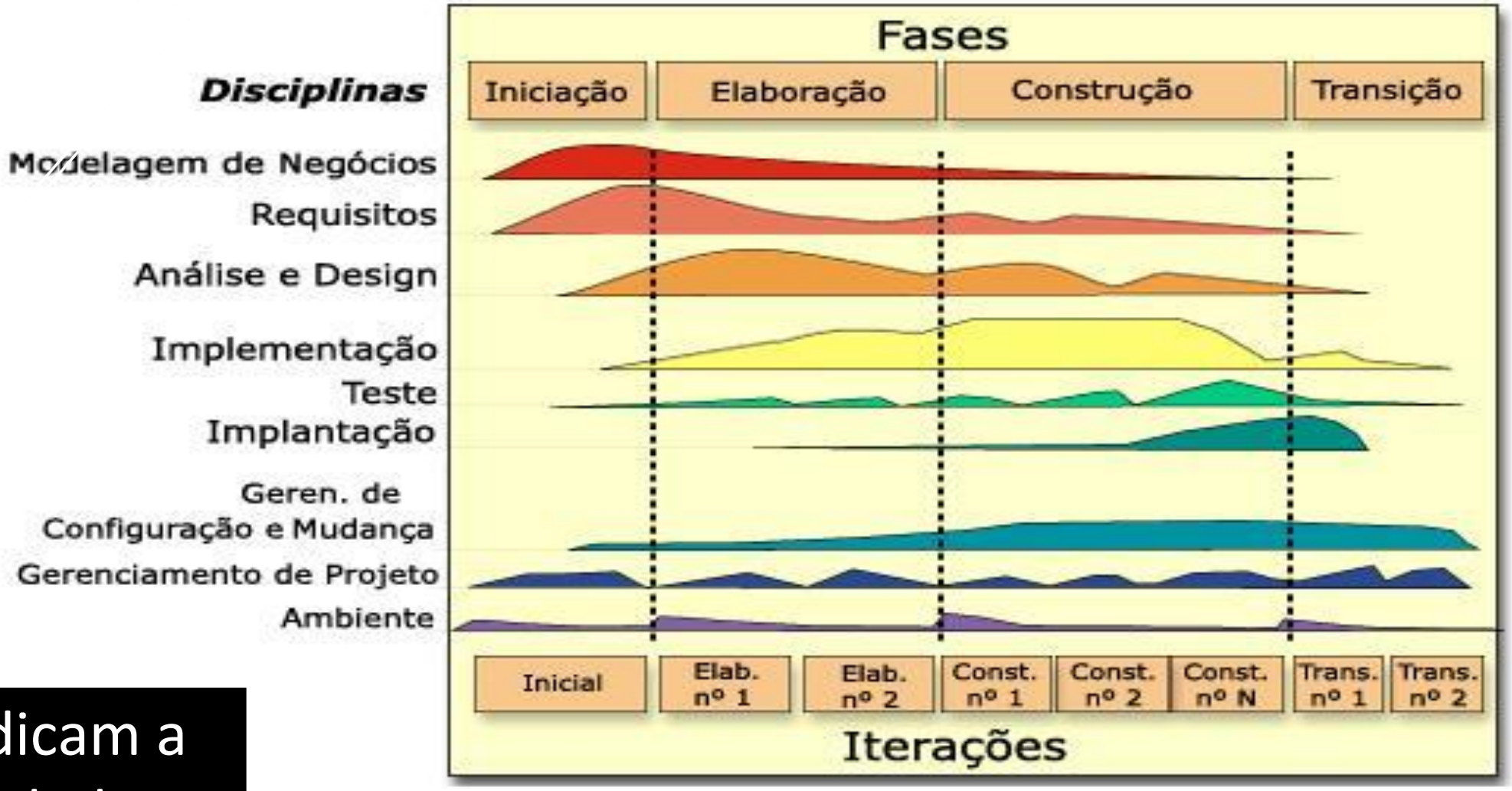
Pressman

FMU
CENTRO UNIVERSITÁRIO

As duas Dimensões do RUP

- **O processo unificado pode ser descrito por duas dimensões, ao longo de dois eixos:**
  - **Eixo horizontal representa o tempo e mostra o aspecto dinâmico do processo. Ele é expresso em termos de ciclos, fases, iterações e marcos. (Fases)**
  - **Eixo vertical representa o aspecto estático do processo. Ele é descrito em termos de atividades, artefatos, papéis e fluxos. (Disciplinas)**

Fases do Processo Unificado



As fases indicam a ênfase que é dada no projeto em um dado instante.

1. **Iniciação ou Concepção: ênfase no escopo do sistema;**
2. **Elaboração: ênfase na arquitetura;**
3. **Construção: ênfase no desenvolvimento;**
4. **Transição: ênfase na implantação.**

FMU
CENTRO UNIVERSITÁRIO

Ficou para estudar em 

- **A seguir nessa apresentação, temos explicações sobre cada uma das disciplinas do RUP.**
- **Essa semana, uma das tarefas para casa é leitura desses próximos slides.**
- **Na sequência, há uma atividade para ser desenvolvida e entregue na próxima aula.**

Disciplinas do RUP

6 Disciplinas de Engenharia de Software

- ✓ **Modelagem de Negócios**
- ✓ **Requisitos**
- ✓ **Análise e Projeto**
- ✓ **Implementação**
- ✓ **Teste**
- ✓ **Implantação**

3 Disciplinas de Apoio / Suporte

- ✓ **Ambiente**
- ✓ **Configuração e Gerência de Mudança**
- ✓ **Gerência de Projeto**

Disciplinas do RUP

Modelagem de Negócios

- **O objetivo é estabelecer uma melhor compreensão e um canal de comunicação entre engenharia de negócios e a engenharia de software.**
  - **Significa que os engenheiros de software devem compreender a estrutura e a dinâmica da empresa alvo (o cliente)**
    - **Atuais problemas na organização e possíveis melhorias.**
  - **Os engenheiros de software também devem garantir um entendimento comum da organização-alvo entre os clientes, usuários finais e desenvolvedores.**

Os processos de negócio são modelados usando Casos de Uso de Negócios

Disciplinas do RUP

Requisitos

- **Esta disciplina explica como levantar pedidos das partes interessadas e transformá-los em um conjunto de requisitos que os produtos funcionam no âmbito do sistema a ser construído e fornecem requisitos detalhados para o que deve fazer o sistema.**

Os agentes que interagem com o sistema são identificados e os Casos de Uso são desenvolvidos para modelar os requisitos do sistema

Disciplinas do RUP

Análise e Projeto

- O objetivo da análise e projeto é mostrar como o sistema vai ser realizado. Provar que o sistema:
  - ✓Executará as tarefas e funções projetadas.
  - ✓Satisfará os requisitos estabelecidos.
  - ✓Será robusto e ameno a mudanças.

Um modelo de projeto é criado e documentado usando modelos de arquitetura, modelos de componentes, modelos de objeto e modelos de sequência.

Disciplinas do RUP

Implementação

- **Os objetivos são:**
  - ✓ **Organizar o código em subsistemas, camadas, componentes, pacotes.**
  - ✓ **Implementar classes e objetos em termos de componentes.**
  - ✓ **Testar unitariamente os componentes desenvolvidos.**
  - ✓ **Integrar resultados, produzindo um sistema executável.**

- ***Os componentes são implementados e estruturados em subsistemas de implementação.***
- ***A geração automática de código com base nos modelos de projetos ajuda a acelerar esse processo.***
- ***Despende-se esforço na organização e reutilização de componentes.***

Disciplinas do RUP

Teste

- O RUP propõe uma abordagem iterativa, o que significa que deve-se testar durante todo o projeto.
  - ✓ Isto permite encontrar defeitos tão cedo quanto possível, o que reduz radicalmente o custo de reparar o defeito.

O teste é um processo iterativo realizado em conjunto com a implementação.

Disciplinas do RUP

Implantação

❑ As diretrizes para a implantação é de produzir com sucesso lançamentos de produtos e entregar o software para seus usuários finais.

Cobre:

✓Produção de releases externos do software
✓Embalagem do software e aplicativos de negócios
✓Distribuição do software
✓Instalação do software e prestação de ajuda
✓Assistência aos usuários.

Uma versão é criada, distribuída aos usuários e instalada no local de trabalho.

Disciplinas do RUP

Configuração e Gerência de Mudança

- **Gerenciamento de configuração: A gestão de configuração é responsável pela estruturação sistemática dos produtos.**
  - ✓ **Artefatos, como documentos e modelos, precisam estar sob controle de versão e essas alterações devem ser visíveis.**
  - ✓ **Ele também mantém o controle de dependências entre artefatos para que todos os artigos relacionados sejam atualizados quando são feitas alterações**
- **Gerenciamento de solicitações de mudança: Durante o processo de desenvolvimento de sistemas com muitos artefatos existem diversas versões.**
- **Gerenciamento de status e medição: As solicitações de mudança são controladas, assim como seus atributos e a causa raiz, prioridade, etc. Esses estados e atributos são armazenados no banco de dados para produzir relatórios úteis sobre o andamento do projeto.**

Disciplinas do RUP

Gerência de Projetos

❑ **No RUP, a Gerência de Projetos concentra-se principalmente em:**

✓**Balancear objetivos conflitantes dos envolvidos, superando problemas e entregando, de forma bem sucedida, um produto que satisfaz a necessidade de clientes e usuários.**

✓**Fornecer um suporte para gerenciar projetos de software e gerenciamento de risco.**

✓**Fornecer diretrizes práticas para planejar, montar a equipe, executar e monitorar os projetos.**

❑ **Esta disciplina do RUP não tenta cobrir todos os aspectos do gerenciamento de projetos.**

✓**Não abrange questões como: Gestão de Pessoas: Custos, Contratos, etc**

Disciplinas do RUP

Ambiente

- **O foco são nas atividades necessárias para configurar o processo para um projeto.**
- **Ele descreve as atividades necessárias para desenvolver as diretrizes de apoio a um projeto.**
- **Tem como finalidade fornecer e garantir o ambiente adequado para a organização, através de ferramentas e processos capazes de suportar as atividades da equipe de desenvolvimento.'**

Essa disciplina está relacionado à disponibilização de ferramentas apropriadas de software para a equipe de desenvolvimento

Os 5 principais Elementos do RUP

Além das disciplinas, o RUP traz outros 4 elementos considerados como principais.

Os 5 elementos principais do RUP:

- **Papéis (perfil)**
- **Atividades**
- **Artefatos**
- **Fluxos de Trabalho**
- **Disciplinas**

Os 5 principais Elementos do RUP

Papéis

- **Define um conjunto de responsabilidades em termos das atividades que esse papel pode realizar.**
  - **Pode ser desempenhado por um indivíduo ou um conjunto de indivíduos trabalhando como uma equipe.**
- **Ele define o comportamento e as responsabilidades de um indivíduo ou grupo num trabalho em equipe.**
- **Os papéis não são indivíduos e nem cargos ou funções → um indivíduo pode ter vários papéis.**

Os 5 principais Elementos do RUP

Papéis

- ***Exemplos*:**

  • **Analista de sistema: coordena o levantamento dos requisitos e a modelagem dos casos de uso, identificando funções do sistema e estabelecendo o escopo deste.**

  • **Projetista: define responsabilidades, operações, atributos, relacionamentos de uma ou mais classes e determina como devem ser ajustadas para serem implementadas no ambiente.**

  • **Projetista de testes: responsável pelo planejamento, projeto, implantação e avaliação de testes, incluindo a geração de plano e modelo, implementando procedimentos para os testes e avaliando a abrangência e profundidade dos testes.**

Os 5 principais Elementos do RUP

Atividades

- **Uma atividade é uma unidade de trabalho que um indivíduo executa quando está exercendo um determinado papel.**
- **É descrita por seus passos e artefatos de entrada e saída.**

- **Exemplos:**

✓**Planejar uma iteração: realizada pelo papel gerente de projeto.**

✓**Encontrar casos de uso e atores: realizada pelo papel analista de sistemas.**

✓**Rever o projeto: realizada pelo papel revisor de projeto.**

✓**Executar um teste de *performance*: realizado pelo papel testador de performance.**

FMU
CENTRO UNIVERSITÁRIO

Os 5 principais Elementos do RUP

Artefatos

- **É um trecho de informação que é produzido, modificado ou utilizado em um dado processo, sendo produzido durante o desenvolvimento deste.**
- **Podem ser usados como entradas de atividades e podem ser produzidos como saída.**
- **Exemplos**

✓**Modelo: como um modelo de caso de uso, um modelo de projeto.**

✓**Elemento de um modelo: como uma classe, um caso de uso, um subsistema.**

✓**Documento: como um caso de negócio, glossário, visão.**

✓**Código fonte.**

✓**Executável.**

FMU
CENTRO UNIVERSITÁRIO

Os 5 principais Elementos do RUP

Fluxos de Trabalho (Workflows)

- **Determina a sequência do desenvolvimento das atividades para que possam ser produzidos artefatos de valor.**
- **É uma sequência de atividades que são executadas para a produção de um resultado. Pode ser representado por diagramas de sequência, diagramas de colaboração e diagramas de atividades da linguagem UML.**

FMU
CENTRO UNIVERSITÁRIO

Os 5 principais Elementos do RUP

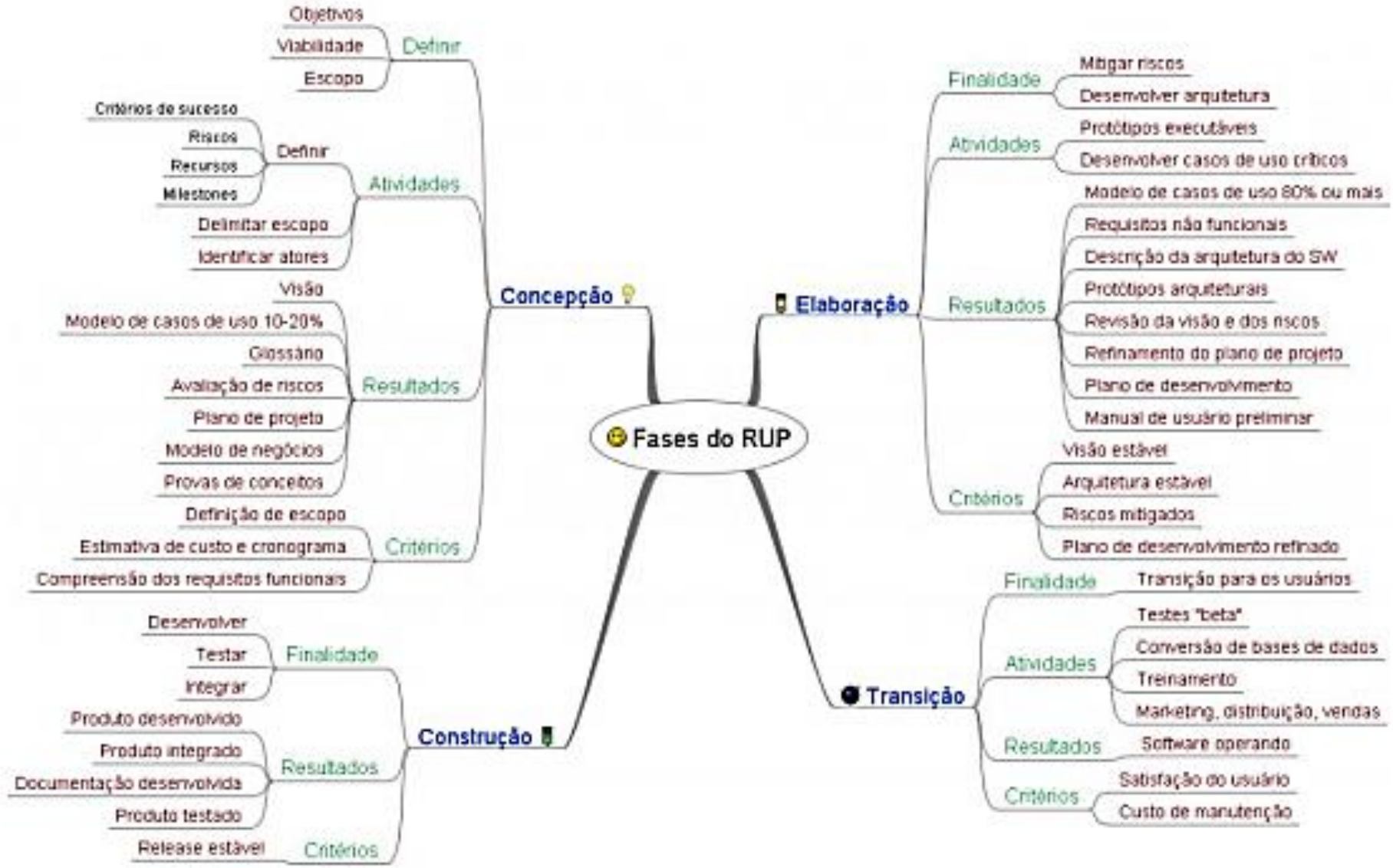
Fluxos de Trabalho (Workflows)

- **O RUP utiliza três tipos de fluxos de trabalho:**

✓**Fluxos de trabalho principais, associados com cada disciplina.**

✓**Fluxos de trabalho de detalhe, para detalhar cada fluxo de trabalho principal.**

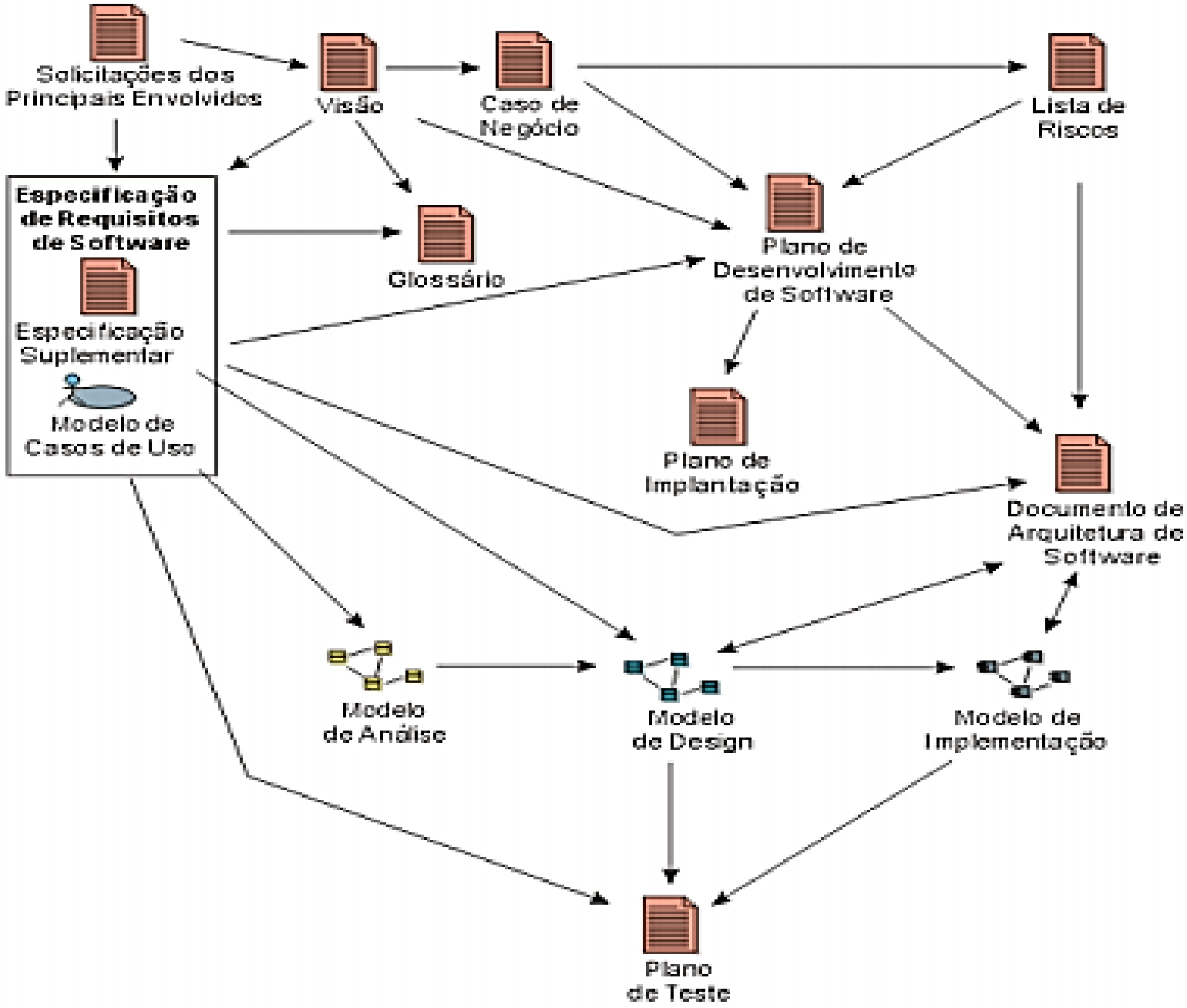
✓**Planos de iteração, que mostram como a iteração deverá ser executada.**



Mapa Mental de Engenharia de Software – Fases do RUP

Fonte: https://www.diegomacedo.com.br/mapa-mental-de-engenharia-de-software-fases-do-rup/

FMU
CENTRO UNIVERSITÁRIO



Os principais artefatos do Rational Unified Process e o fluxo de informações existente entre eles

FMU
CENTRO UNIVERSITÁRIO

ATIVIDADE 3 – para ser entregue 23/03/20
Em grupos de até 4 alunos

Tema: **ciclo de desenvolvimento de software – modelo RUP.**

Imaginem que fazem parte de uma equipe de desenvolvimento que agora está adotando a metodologia RUP.

Ficou para entregar hoje...

1. **Qual o software que a equipe de vocês irá desenvolver?**
2. **Como RUP será utilizado?**

Deverão ser descritas as fases que serão aplicadas do RUP e quais serão os principais artefatos produzidos em cada fase.

3. **Qual será a estratégia adotada?**

Deverão ser descritos como as iterações serão aplicadas (módulos a serem produzidos em cada uma) e quais os papeis serão necessários para o projeto e como serão distribuídos pelo time.

4. **Quais vantagens e desvantagens ao utilizar esse ciclo?**
5. **Quais os possíveis riscos e como mitigá-los?**

Vamos conferir se você lembra os pontos principais da aula passada.

Quais são os principais pontos do manifesto ágil?

- **Desenvolvendo e ajudando outros a desenvolver software, estamos desvendando formas melhores de desenvolvimento.**

Por meio deste trabalho passamos a valorizar:

- ***Indivíduos e interações acima de processos e ferramentas***
- ***Software operacional acima de documentação completa***
- ***Colaboração dos clientes acima de negociação contratual***
- ***Respostas a mudanças acima de seguir um plano***

FMU
CENTRO UNIVERSITÁRIO

Não lembra do manifesto ágil???

Dê uma revisada nos slides 65 e 66 da aula passada (aula 4)

Vamos revisar o que de principal a filosofia ágil defende e seus valores!

O processo de Software baseado em Metodologias Ágeis

❑ A filosofia defende:

- ✓A satisfação do cliente e a entrega de incremental prévio;
- ✓Equipes de projeto pequenas e altamente motivadas;
- ✓Métodos informais;
- ✓Artefato de engenharia de software mínimos e, acima de tudo, simplicidade no desenvolvimento geral.

❑ Os princípios de desenvolvimento:

- ✓Priorizam a entrega mais que análise e projeto.
- ✓Também priorizam a comunicação ativa e contínua entre desenvolvedores e clientes.

FMU
CENTRO UNIVERSITÁRIO

Os valores do Manifesto Ágil:

- **INDIVÍDUOS E INTERAÇÃO** mais que Ferramentas e Processos
- **SOFTWARE FUNCIONANDO** mais que Documentação Abrangente
- **COLABORAÇÃO COM CLIENTE** mais que Negociação de Contratos
- **RESPONDER A MUDANÇAS** mais que Seguir um Plano

FMU
CENTRO UNIVERSITÁRIO

O Processo Ágil

- Vários modelos de processo foram propostos seguindo os princípios das metodologias ágeis, como XP e Scrum.

Extreme Programming Project

FMU
CENTRO UNIVERSITÁRIO

eXtreme Programming

- É considerada a abordagem mais amplamente utilizada para o desenvolvimento de software ágil.
- 5 Valores Básicos da XP
  - ✓Comunicação
  - ✓Simplicidade
  - ✓Feedback
  - ✓Coragem
  - ✓Respeito

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming - Valores Básicos

1. Comunicação

A XP enfatiza a colaboração estreita, embora informal (verbal), ENTRE CLIENTES E DESENVOLVEDORES, o estabelecimento de metáforas eficazes para comunicar conceitos importantes, FEEDBACK (realimentação) contínuo e evitar documentação volumosa como meio de comunicação.

"Os problemas nos projetos invariavelmente recaem sobre alguém não falando com alguém sobre algo importante"

Kent Beck

eXteme Programming - Valores Básicos

Metáforas são muito usadas na XP onde os desenvolvedores criam elementos dentro do computador para simular outros do mundo real.

A lixeira, a mesa de trabalho, janelas, pastas e outros itens que estamos habituados a encontrar no computador, simulam elementos do mundo físico e seus respectivos comportamentos.

A XP procura explorar a utilização de metáforas, para que clientes e desenvolvedores estabeleçam um vocabulário apropriado para o projetode modo a elevar a compreensão mútua.

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming - Valores Básicos

2. Simplicidade

A XP restringe os desenvolvedores a projetar apenas para as necessidades imediatas, em vez de considerarem as necessidades futuras.

O intuito é criar um projeto simples que possa ser facilmente implementado em código.

Se o projeto tiver que ser melhorado, ele poderá ser refatorado mais tarde.

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming - Valores Básicos

3. Feedback

provém de três fontes:

a) do próprio software implementado,

b) do cliente e

c) de outros membros da equipe de software.

Através da elaboração do projeto e da implementação de uma estratégia de testes eficaz, o software (via resultados de testes) propicia um feedback para a equipe ágil.

A XP faz uso do teste de unidades como sua tática de testes primária.

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming - Valores Básicos

4. Coragem

Kent Beck afirma que a adoção estrita a certas práticas da XP exige coragem.

Coragem, nesse caso, significa disciplina.

Quando há uma pressão significativa para a elaboração do projeto pensando em futuros requisitos...

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming - Valores Básicos

... a maioria das equipes de software sucumbe, argumentando que "projetar para amanhã" poupará tempo e esforço no longo prazo

... já uma equipe XP ágil deve ter disciplina (coragem) para projetar para hoje, reconhecendo que as necessidades futuras podem mudar dramaticamente exigindo, consequentemente, substancial retrabalho em relação ao projeto e ao código implementado.

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming - Valores Básicos

5. Respeito

É um valor que dá sustentação a todos os demais.

Membros de uma equipe só irão se preocupar em comunicar-se melhor, por exemplo, se importarem uns com os outros.

Respeito é o mais básico de todos os valores!

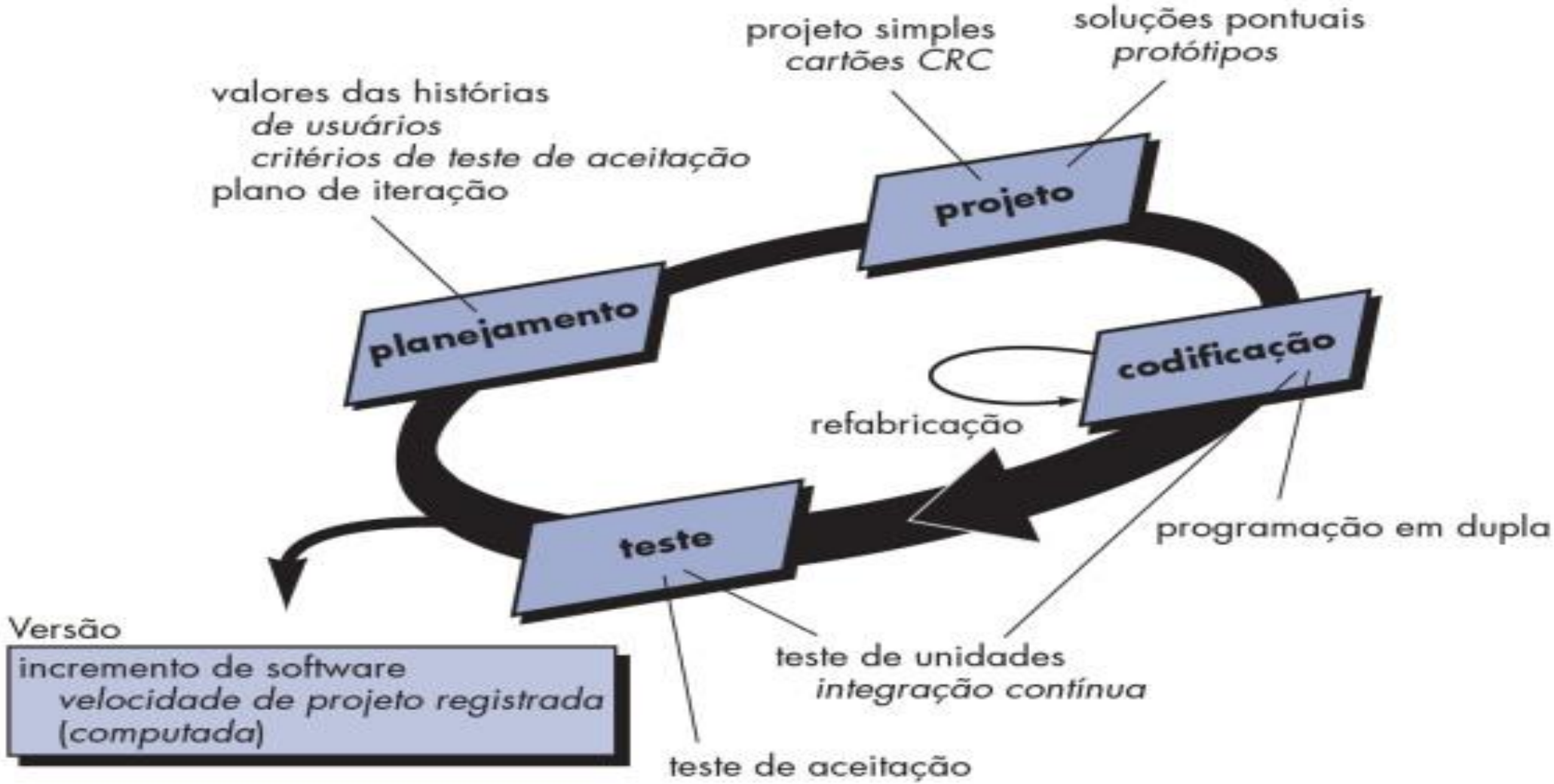
Se ele não existir em um projeto, não há nada que possa salvá-lo!

Saber ouvir, saber compreender e respeitar o ponto de vista do outro é essencial para que um projeto de software seja bem sucedido!

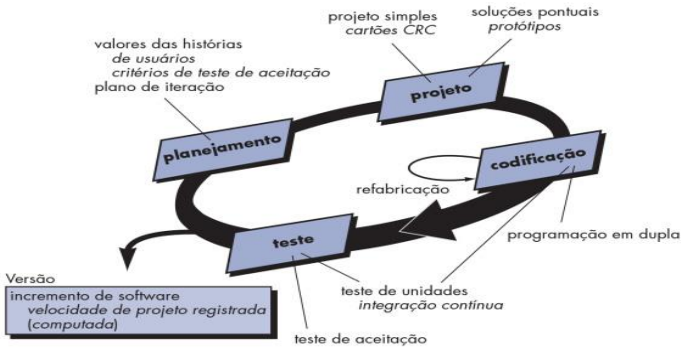
eXteme Programming
O Processo

- ✓ Abordagem orientada a objetos como paradigma de desenvolvimento preferido.
- ✓ Envolve um conjunto de regras e práticas constantes no contexto de quatro atividades metodológicas:
  1. Planejamento,
  2. Projeto,
  3. Codificação e
  4. Testes

eXteme Programming
O Processo



XP – Fases: Planejamento



❑ Ouvir

⇒ conduz à criação de um conjunto de "histórias" (*histórias de usuários)*

***Histórias de Usuários**: descreve o resultado, as características e a funcionalidade requisitados* para o software a ser construído.

O Jogo do planejamento: se inicia com a atividade de ouvir...

É uma atividade de levantamento de requisitos que capacita os membros técnicos da equipe XP a entender o ambiente de negócios do software e possibilita que se consiga ter uma percepção ampla sobre os resultados solicitados, fatores principais e funcionalidades.

FMU
CENTRO UNIVERSITÁRIO

Uma Estória do Usuário descreve um detalhamento de alto nível de uma funcionalidade e/ou de um item do Product Backlog. E facilita na estimava da velocidade da esquipe

	Como cliente, eu quero fazer reserva de apartamento pela web para
	facilitar o planejamento das minhas férias

O Caso de Uso especificam a interação entre o Usuário e o Sistema.

FMU
CENTRO UNIVERSITÁRIO

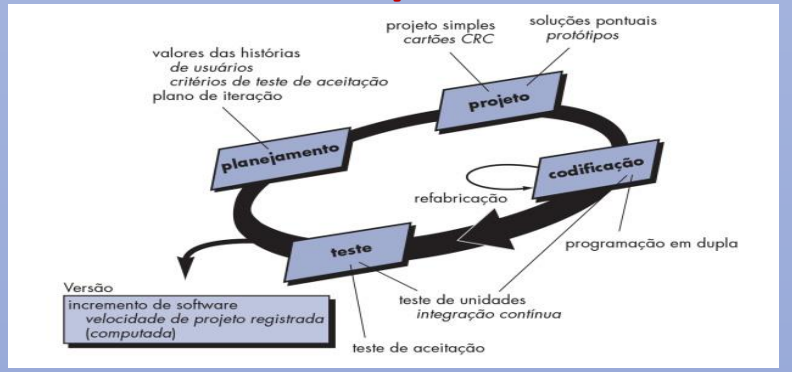
	Como cliente, eu quero sacar dinheiro em qualquer caixa eletrônico
	para que não tenha que ir na agência bancária.

	Como paciente, eu quero agendar minha consulta médica pela web
	para que não tenha que usar o telefone.

XP – Fases: Planejamento

Clientes + desenvolvedores trabalham juntos

✓ Decidem como agrupar histórias para a versão seguinte (próximo incremento) a ser desenvolvida pela equipe XP.



✓Chega-se a um compromisso básico

- quais histórias serão incluídas,
- data de entrega e
- outras questões de projeto

✓ A equipe XP ordena as histórias a serem desenvolvidas em uma das três formas:

1) Todas serão implementadas imediatamente – em um prazo de poucas semanas,
2) As histórias de maior valor serão deslocadas para cima no cronograma e implementadas primeiro ou
3) As histórias de maior risco serão deslocadas para cima no cronograma e implementadas primeiro.

XP – Fases: Planejamento

✓Primeira versão (incremento) entregue...

... a equipe calcula a velocidade do projeto de forma simples...

... a velocidade do projeto é o número de histórias de clientes implementadas durante a primeira versão.

Dessa forma:

- É possível calcular as datas para as próximas entregas e o cronograma do projeto;
- Determinar se foi assumido um compromisso exagerado.

FMU
CENTRO UNIVERSITÁRIO

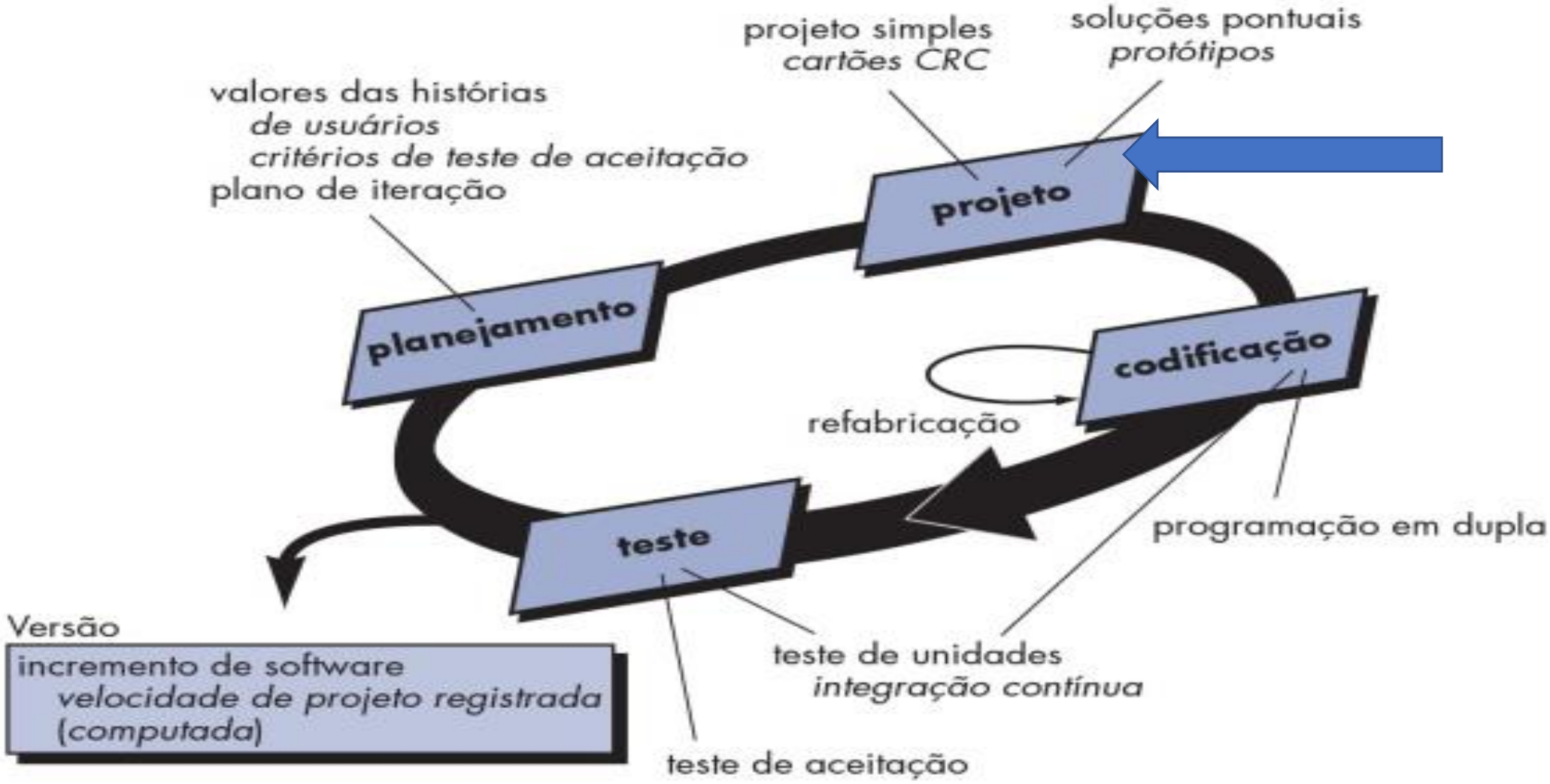
XP – Fases: Planejamento

Conforme o trabalho de desenvolvimento prossegue...
...o cliente pode acrescentar histórias...
...mudar o valor de uma existente...
...dividir algumas ou eliminá-las...
...em seguida...

...a equipe XP reconsidera todas as versões remanescentes

e modifica seus planos.

eXteme Programming
O Processo



FMU
CENTRO UNIVERSITÁRIO

XP – Fases: Projeto

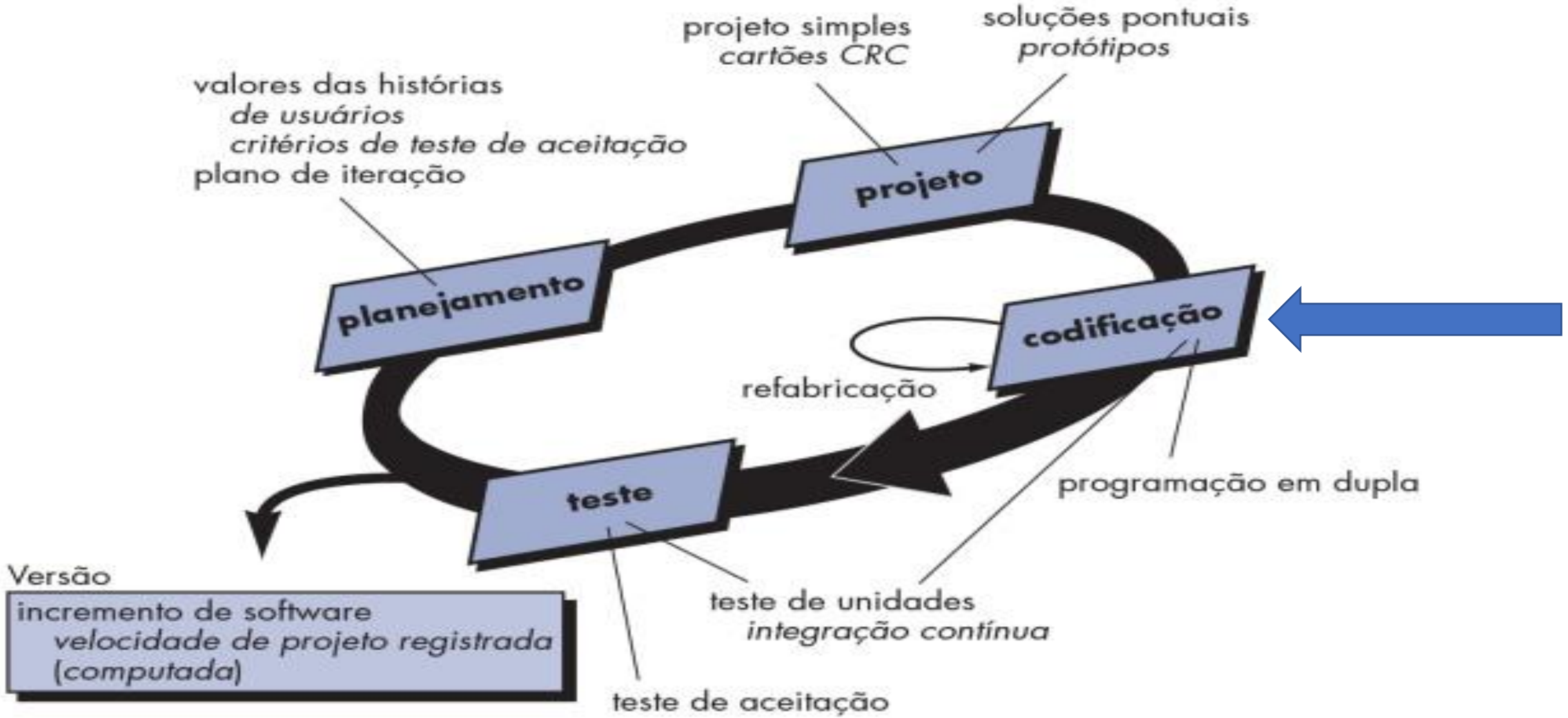
✓Princípio KIS ➔ Keep It Simple preserve a simplicidade

✓Se um difícil problema de projeto for encontrado como parte do projeto de uma história, a XP recomenda a criação imediata de um protótipo operacional dessa parte do projeto.

⇒ Como em outros modelos de processos de desenvolvimento de software, é a fase de design (modelagem), transformando os requisitos (histórias) em documentos que o programador consegue entender.

➢ Só que seguindo o princípio KIS.

eXteme Programming
O Processo



XP – Fases: Codificação

✓Conceito Chave: Programação em Pares

✓A XP recomenda que duas pessoas trabalhem juntas em uma mesma estação de trabalho para criar código para uma história.

✓ Isso fornece um mecanismo para resolução de problemas em tempo real → duas cabeças normalmente funcionam melhor do que uma ☺ e garantia da qualidade em tempo real na medida em que o código é revisto à medida que é criado

✓ Ele também mantém os desenvolvedores focados no problema em questão.

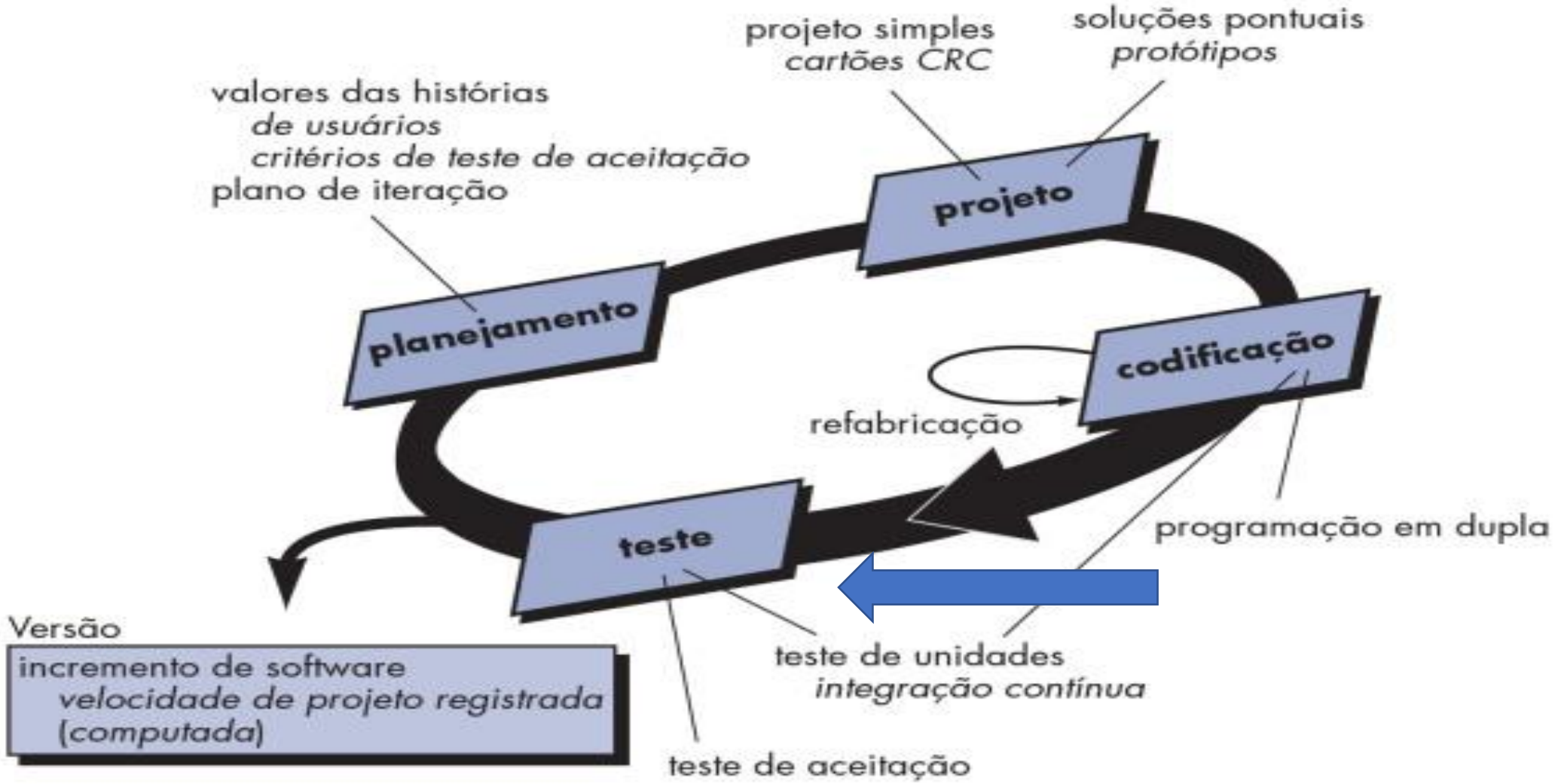
✓ Na prática, cada pessoa assume um papel ligeiramente diferente.

- ❖ Por exemplo, uma pessoa poderia pensar nos detalhes de codificação de determinada parte do projeto...

  ... enquanto outra assegura que padrões de codificação (uma parte exigida pela XP) sejam seguidos ou que o código para a história passará no teste de unidades desenvolvido para validação do código em relação à história.

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming
O Processo



XP – Fases: Testes

- ✓ **Antes de começar a codificar, são feitos testes unitários.**
  - **Exercita-se cada uma das histórias a serem incluídas na versão corrente.**
- ✓ **Os testes de unidade criados devem ser implementados usando-se uma metodologia que os capacite a ser automatizados (podendo ser executados fácil e repetidamente).**

Cada vez que um novo incremento é acrescentado, **n**ovos caminhos de fluxo de dados são estabelecidos - podem ocorrer novas entradas e saídas e nova lógica de controle é chamada.

teste de regressão *é a reexecução do mesmo subconjunto de* testes que já foram executados para assegurar que as alterações não tenham propagado efeitos colaterais indesejados.

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming - RESUMO

- As iterações são curtas. As mudanças devem ser incrementais e feitas aos poucos.
- Os planejamentos de release e das iterações são feitos com base na história dos usuários e conta com a colaboração de toda a equipe de desenvolvimento, inclusive o cliente.
- Tais histórias descrevem cenários com situações de utilização que os envolvidos gostariam que o sistema viesse a oferecer.
- A princípio, devem ser escritas pelos próprios usuários.
- No XP, elas substituem longos documentos de requisitos dos métodos tradicionais. São a base para a criação de estimativas de tempo que serão utilizadas nas reuniões de planejamento de entregas (*releases*).

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming - RESUMO

- Cada história deve levar de 2 a 3 semanas para ser implementada em uma iteração individual.
- Se levar mais do que isso, é necessário que seja dividida em duas ou mais.
- No final de um release, é feita uma determinação rápida do escopo do próximo, através da combinação de estimativas e prioridades do negócio.
- Um release consiste de várias iterações e, em cada iteração, várias histórias são implementadas.
- Os programadores estimam cada história e dizem quantas eles podem implementar no final do release.

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming - RESUMO

- Os clientes escolhem as principais histórias, que serão implementadas - as que agregam > valor p/negócio.
- No XP, faz-se uso de metáforas, com a intenção de oferecer uma visão geral do sistema de uma forma simples e que possa ser compartilhada por clientes e programadores.
- Utiliza-se a refatoração com constantes melhorias no projeto do software para se adaptar a mudanças.
- Semana de trabalho dos envolvidos é de 40 horas - o cansaço e a insatisfação de trabalhar horas extras pode levar a uma queda da qualidade do código.

FMU
CENTRO UNIVERSITÁRIO

eXteme Programming - RESUMO

- Esta metodologia prega a utilização de padrões de codificação, ou seja, que todos os programadores programem da mesma forma, facilitando o entendimento do código e as alterações realizadas. Fortificando o princípio da propriedade coletiva do código.
- Basicamente, um projeto XP passa pelas fases de exploração, planejamento inicial, iterações do release, produção, manutenção e morte.