Centro Universitário FMU

Disciplina:

Engenharia de Software I
Aula 4
Bacharelado em Ciência da Computação
Tecnologia em Análise e Desenvolvimento de Sistemas

Prof.: Celso Eduardo Guimarães

celso.guimaraes@fmu.br

FMU
CENTRO UNIVERSITÁRIO

ATIVIDADE 2 – 09/03/2020
Em grupos de até 4 alunos

Ficou para entregar hoje...

Tema: **ciclo de desenvolvimento de software.**

Você deseja abrir uma empresa e lançar no mercado um software inovador.

◎ Qual ciclo de vida você irá utilizar como base para o processo de desenvolvimento de seu software?
(Descreva um contexto para sua empresa e as características do tipo do ciclo de vida adotado e o motivo).

◎ Quais são os maiores riscos que você está se expondo?
(Descreva os possíveis problemas de adotar o ciclo de vida escolhido).

Agora cada grupo vai apresentar de forma rápida e objetiva sua solução para a sala

FMU
CENTRO UNIVERSITÁRIO

Vamos pensar nessas questões...

Como o ciclo de vida auxilia no desenvolvimento e manutenção de software?

FMU
CENTRO UNIVERSITÁRIO

Vamos pensar nessas questões...

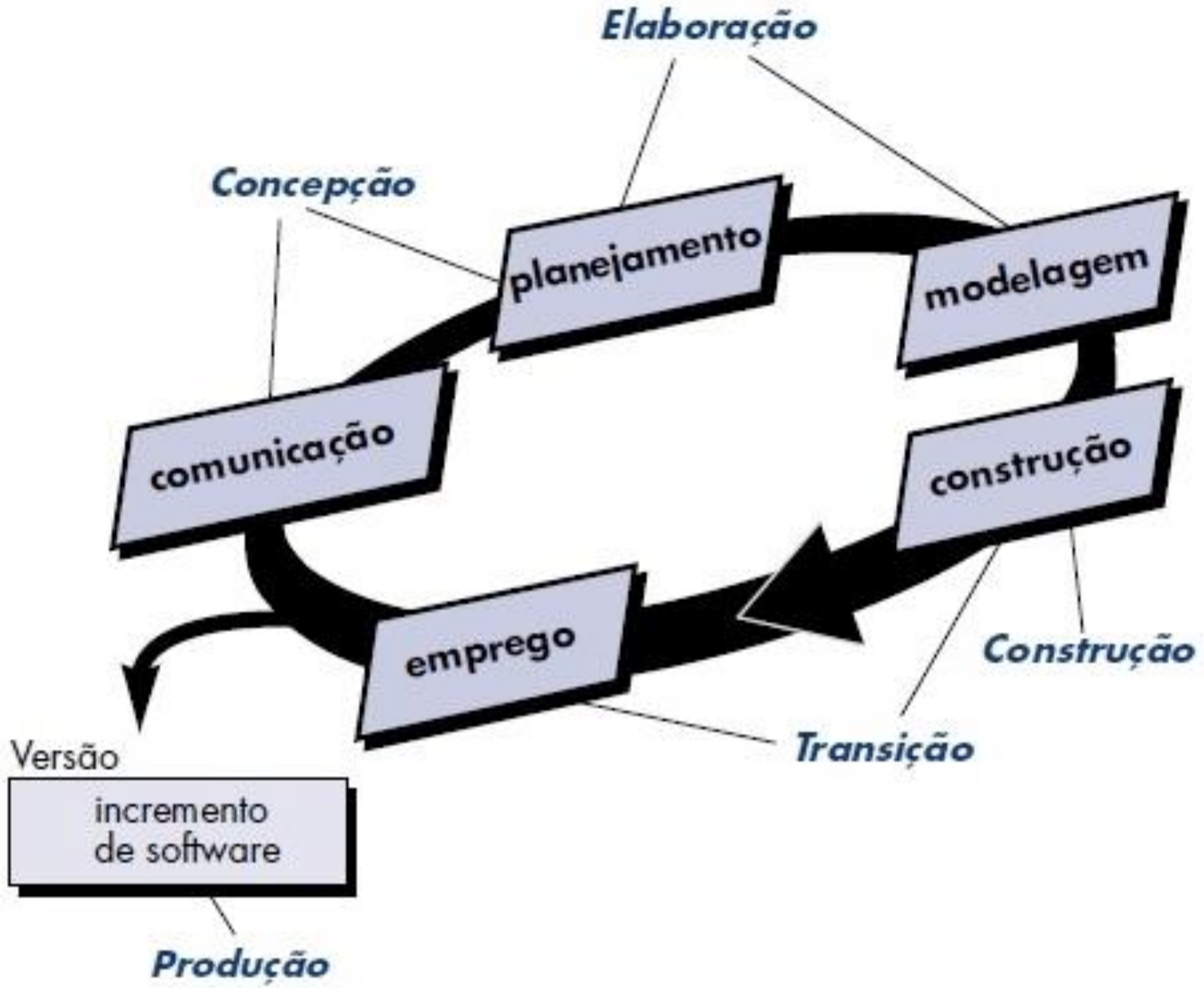
O que pode dar errado em cada etapa do ciclo de desenvolvimento de software?

FMU
CENTRO UNIVERSITÁRIO

https://www.youtube.com/watch?v=6_fVcpC0cxE

FMU
CENTRO UNIVERSITÁRIO

Vamos agora estudar um dos principais modelos de desenvolvimento de software.
Ele é popularmente conhecido como RUP.
Sigla que significa: Rational Unified Process


Elaboração
Concepção
planejamento
modelagem
comunicação
construção
emprego
Construção
Transição
Versão
incremento
de software
Produção

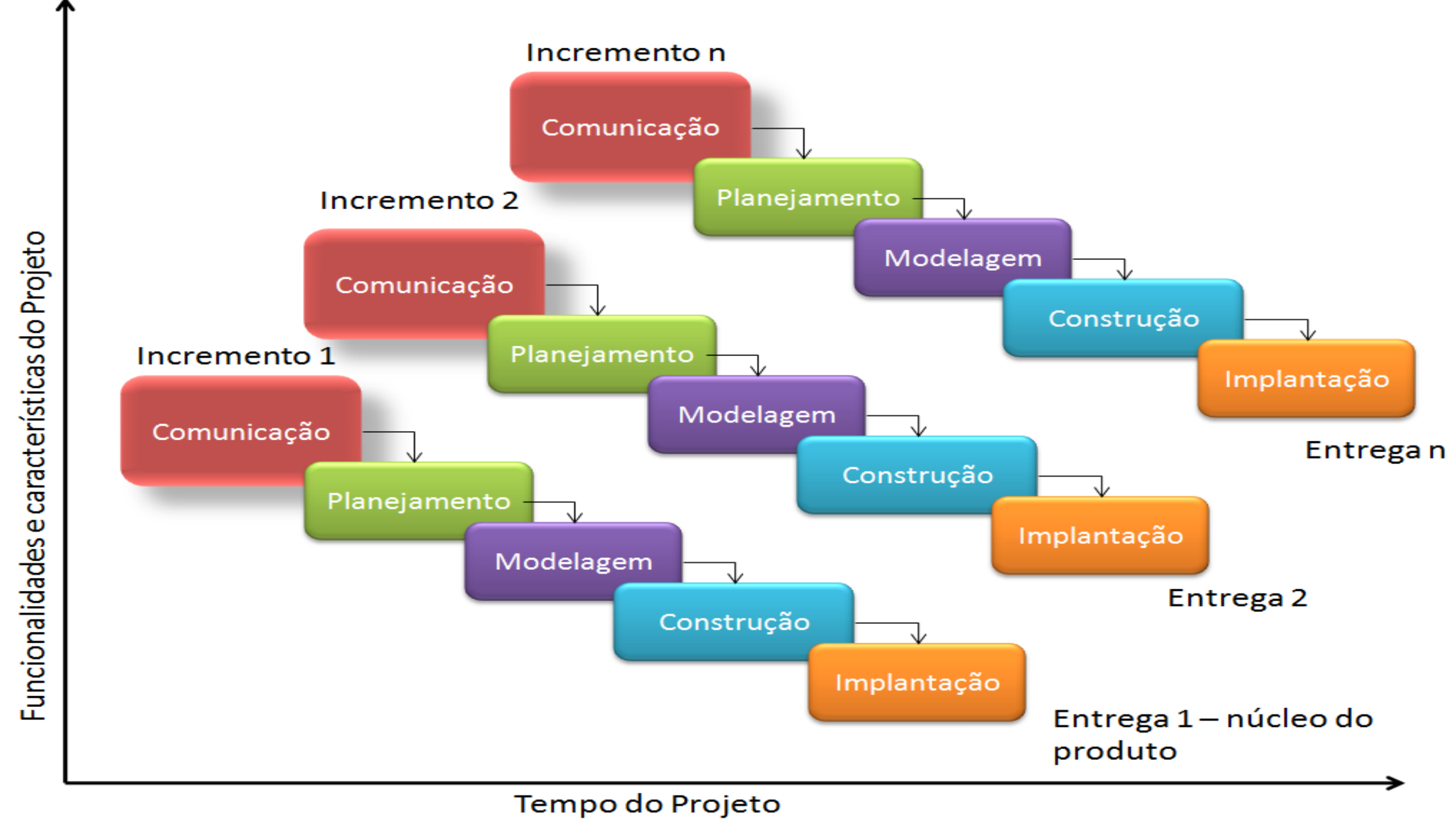
FMU
CENTRO UNIVERSITÁRIO

Processo Unificado

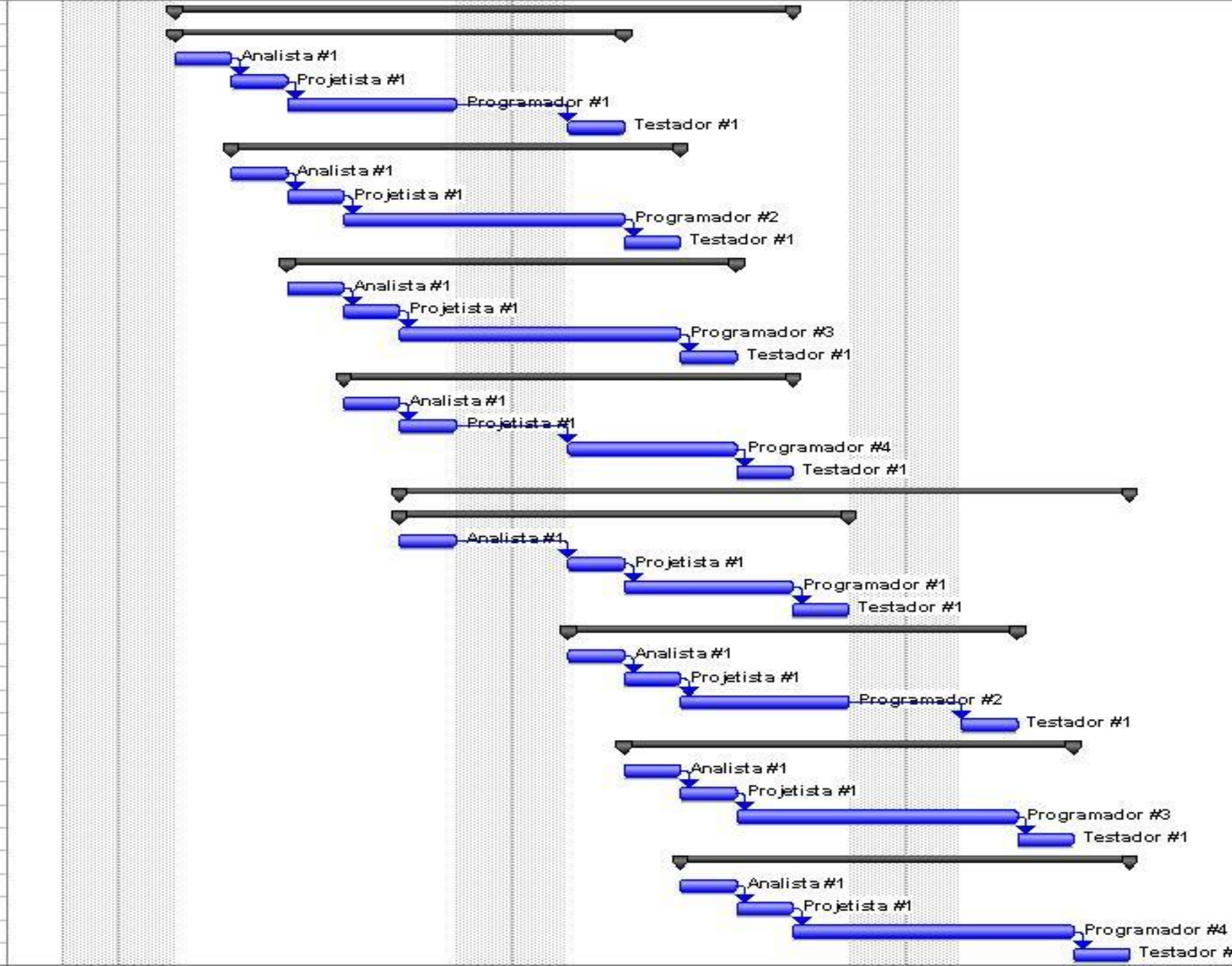
- É um modelo de processo de software baseado no modelo incremental, visando a construção de software orientado a objetos.

- Usa como notação de apoio a **UML**

  → *Unified Modeling Language*


Incremento n
Comunicação
Planejamento
Modelagem
Construção
Implantação
Entrega n
Incremento 2
Comunicação
Planejamento
Modelagem
Construção
Implantação
Entrega 2
Incremento 1
Comunicação
Planejamento
Modelagem
Construção
Implantação
Entrega 1 – núcleo do produto
Funcionalidades e características do Projeto
Tempo do Projeto

Release #1	
Funcionalidade #1	
Especificação	Analista #1
Projeto	Projetista #1
Programação	Programador #1
Testes	Testador #1
Funcionalidade #2	
Especificação	Analista #1
Projeto	Projetista #1
Programação	Programador #2
Testes	Testador #1
Funcionalidade #3	
Especificação	Analista #1
Projeto	Projetista #1
Programação	Programador #3
Testes	Testador #1
Funcionalidade #4	
Especificação	Analista #1
Projeto	Projetista #1
Programação	Programador #4
Testes	Testador #1
Release #2	
Funcionalidade #5	
Especificação	Analista #1
Projeto	Projetista #1
Programação	Programador #1
Testes	Testador #1
Funcionalidade #6	
Especificação	Analista #1
Projeto	Projetista #1
Programação	Programador #2
Testes	Testador #1
Funcionalidade #7	
Especificação	Analista #1
Projeto	Projetista #1
Programação	Programador #3
Testes	Testador #1
Funcionalidade #8	
Especificação	Analista #1
Projeto	Projetista #1
Programação	Programador #4
Testes	Testador #1



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

Mas ainda não é hora de explorarmos a UML.
Vamos focar no modelo RUP...

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.**

FMU
CENTRO UNIVERSITÁRIO

Processo Unificado

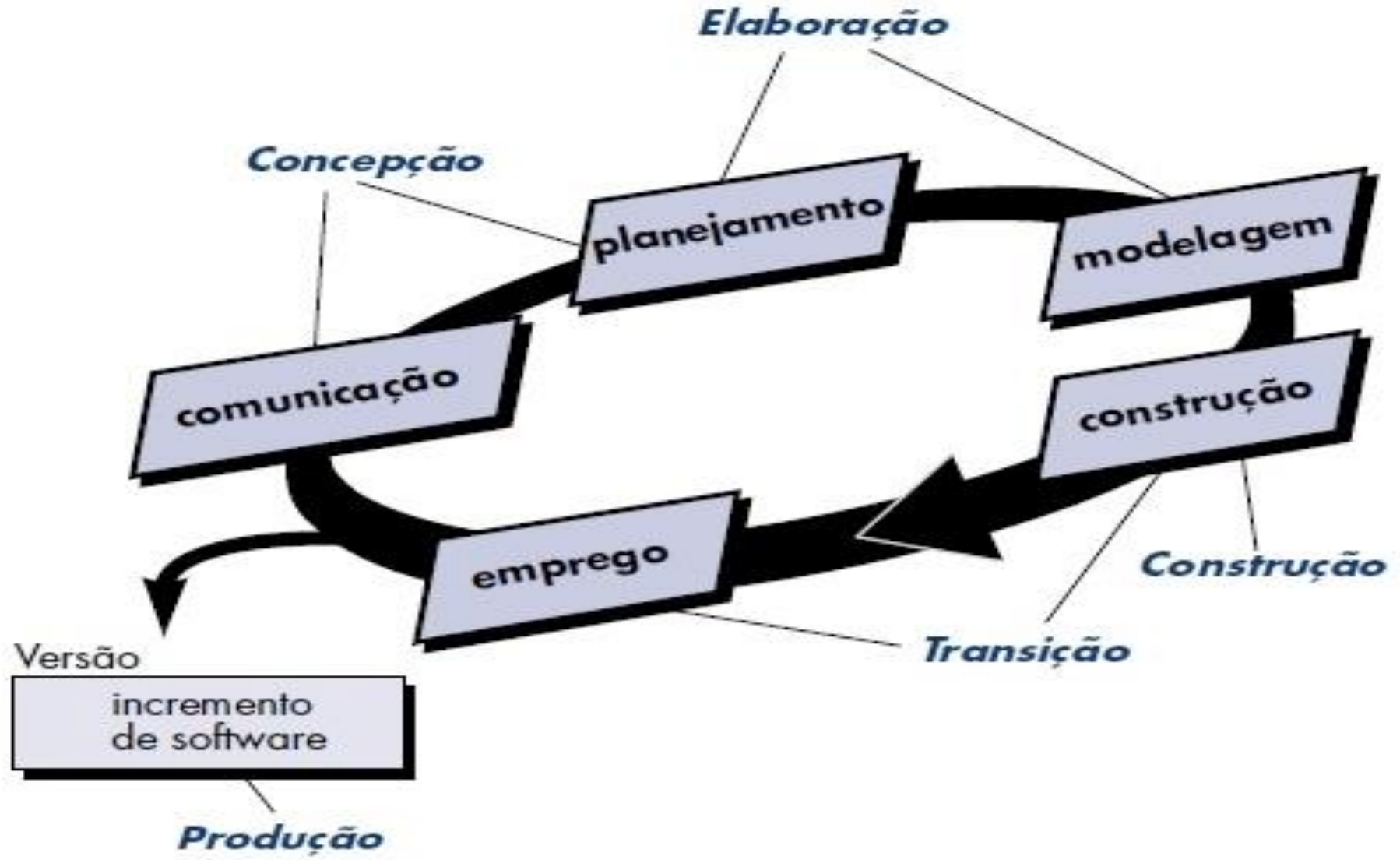
- **O processo é uma tentativa de aproveitar as melhores práticas e as melhores características dos processos tradicionais.**

- Dedica grande foco em
  - ✓ **Dar importância à comunicação com o cliente;**
  - ✓ **Métodos racionalizados (sequenciados) para descrever a visão do cliente sobre um sistema – Casos de Uso;**
  - ✓ **Enfatizar a importância do papel da arquitetura de software (ajuda o arquiteto a manter o foco nas metas corretas: compreensibilidade, confiança em mudanças futuras e reutilização);**
  - ✓ **Sugerir um fluxo de processo iterativo e incremental, proporcionando a sensação evolucionária.**

Rational Unified Process

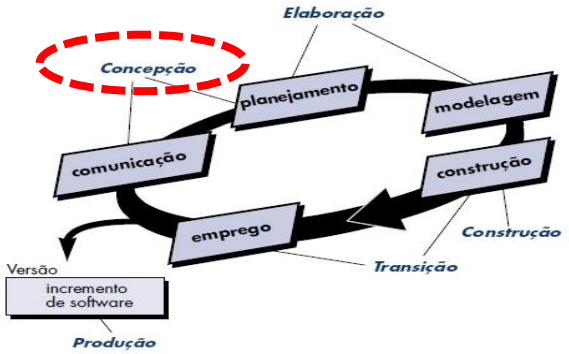
- **É suportado por ferramentas que automatizam grande parte das atividades de desenvolvimento.**
- **É processo configurável.**
- **Dificilmente um único processo é adequado para o desenvolvimento de todos os tipos de software.**
- **Ele pode ser ajustado (customizado) para pequenas equipes de desenvolvimento bem como para grandes organizações.**
- **Captura as melhores práticas de desenvolvimento da Engenharia de Software.**

Fases do Processo Unificado



Pressman

FMU
CENTRO UNIVERSITÁRIO

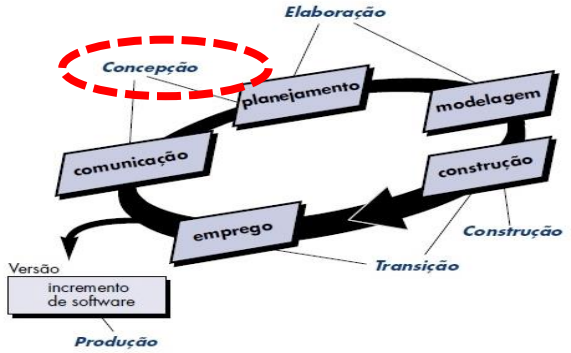


Concepção

- O objetivo dessa fase é estabelecer um Business Case para o sistema.
- Identifica-se todas as Entidades Externas (inclui pessoas e sistemas) que irão interagir com o sistema a ser construído, definindo também suas interações.

- As informações são usadas para avaliar a contribuição do sistema com o negócios para entendimento da viabilidade do sistema (deve-se seguir em frente com o projeto?)

Fases do Processo Unificado



Concepção

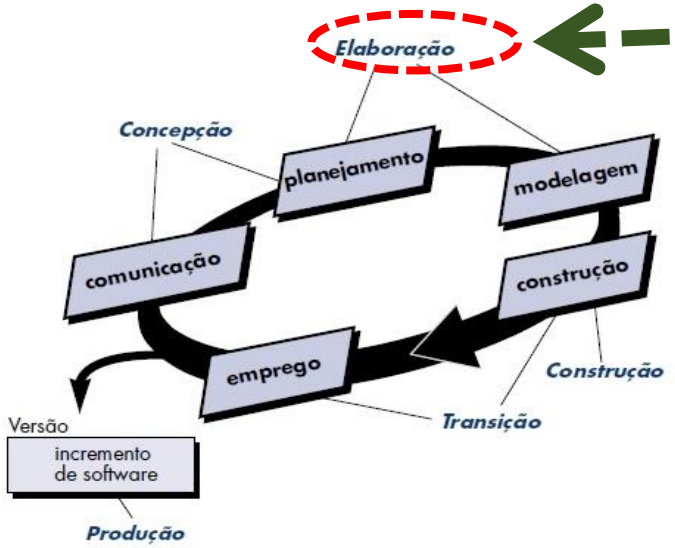
- Protótipo pode ser desenvolvido para validação do cliente.

- As iterações devem ser definidas quanto: quantidade / objetivos.

- O planejamento identifica RECURSOS, AVALIA RISCOS, define um CRONOGRAMA e estabelece uma base para as fases que serão aplicadas à medida que o incremento de software é desenvolvido.

Fases do Processo Unificado

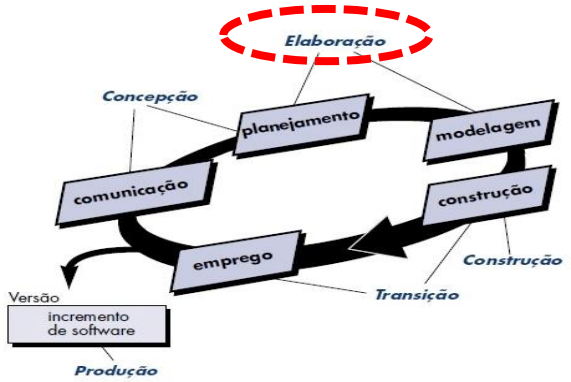
Elaboração



- **OBJETIVO**:
  - ✓ Um modelo de requisitos para o sistema;
  - ✓ Uma descrição de arquitetura;
  - ✓ Um plano de desenvolvimento para o software.

- Essa fase envolve atividades de planejamento e modelagem.
- Nela devem ser desenvolvidas os casos práticos iniciais levantados na fase anterior (concepção).
- Nessa fase é essencial chegar a um entendimento do domínio do problema.

Fases do Processo Unificado



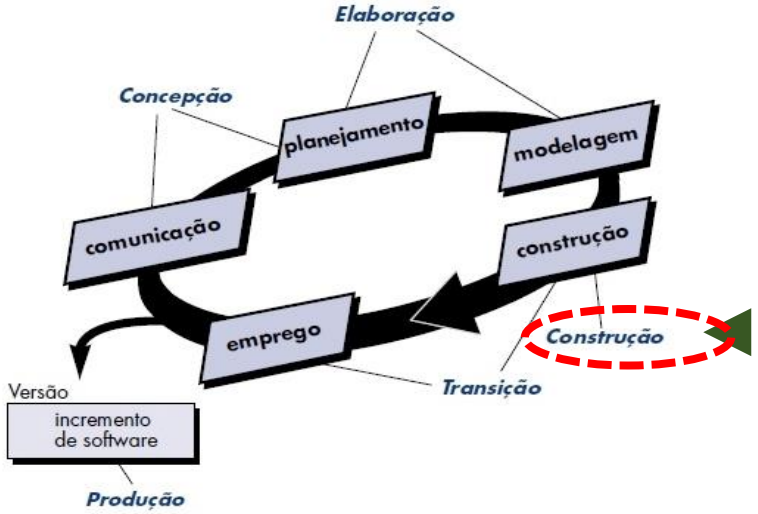
Elaboração

- Essa etapa amplia a representação da arquitetura para o sistema.

- Deve buscar complementar o levantamento (documentação dos casos de uso), voltado para a arquitetura do sistema.
- Revisar a modelagem do negócio para os projetos.

- Deve buscar o entendimento de:
  - -O plano do projeto é confiável?
  - - Custos são admissíveis?

Fases do Processo Unificado



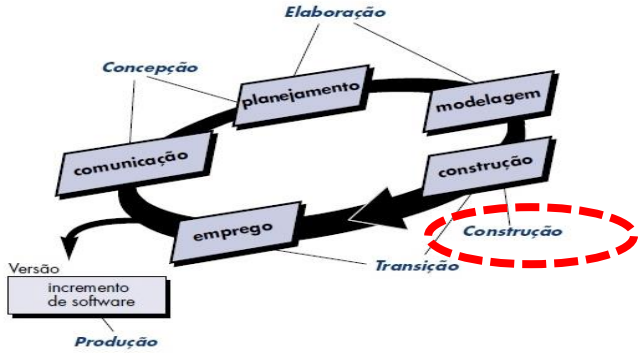
Construção

- Relacionada principalmente às atividades de programação e testes do sistema.

- As partes do sistema são desenvolvidas paralelamente e integradas durante essa fase do PU.

- Ao concluir essa fase:
  - Deve-se ter um sistema de software em funcionamento e a documentação associada pronta para ser liberada para os usuários do sistema.

Fases do Processo Unificado

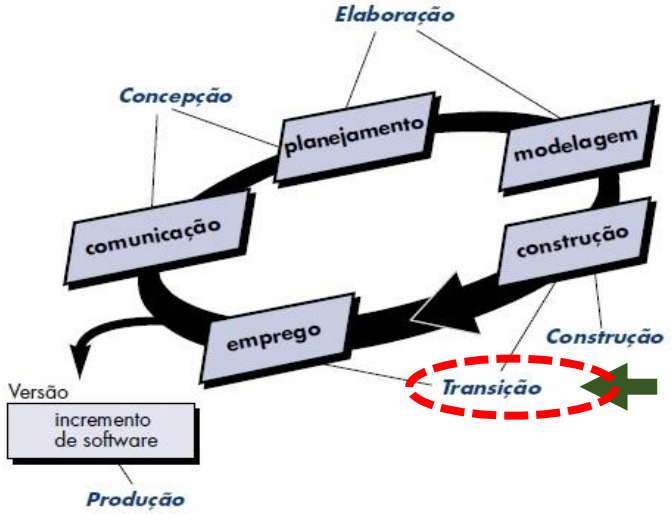


Construção

- A fase de construção do PU é idêntica à atividade de construção definida para o processo de software genérico.

- A entrada para a fase de construção é Modelo de Arquitetura.
- A fase de construção desenvolve ou adquire componentes de software.
- Esses componentes farão com que cada caso prático se torne operacional para os usuários finais.
- Os modelos de requisitos e de projeto (iniciados na fase de elaboração), são completados para refletir a versão final do incremento de software.

Fases do Processo Unificado

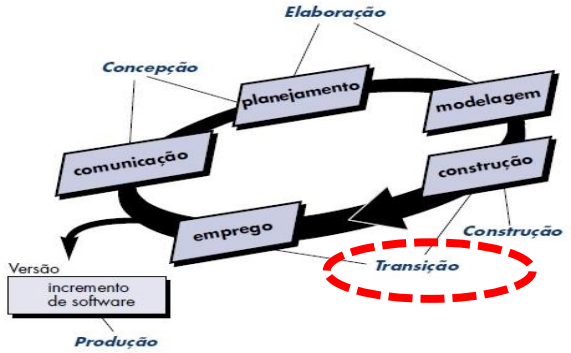


Transição

- Transferência do sistema do "Ambiente de Desenvolvimento" para o "Ambiente de Produção".
- Go Live no ambiente real!

- Ao final dessa fase:
  - -Deverá ter um sistema de software
    - Documentado
    - Funcionando Corretamente
  - - Documentação e Software devem ser disponibilizados no ambiente operacional.

Fases do Processo Unificado

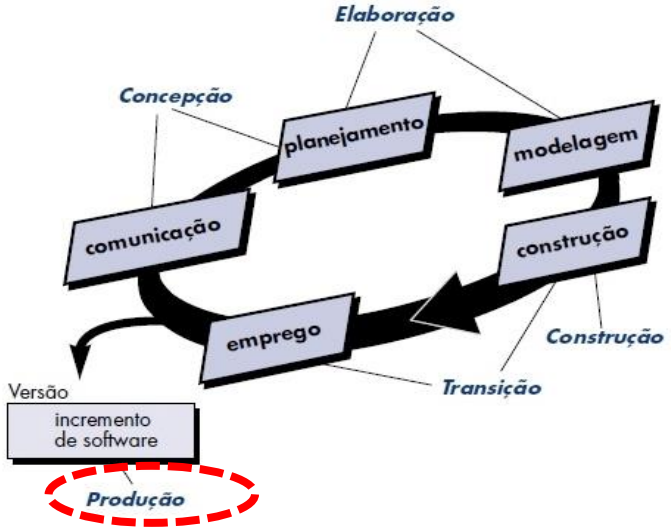


Transição

- Essa fase ainda possui os últimos estágios da Construção.
- Entrega-se o software aos usuários finais para testes beta e o feedback dos usuários relata defeitos e mudanças necessárias.

- Além disso, a equipe de software elabora material com as informações de apoio (por exemplo, manuais para o usuário, guias para resolução de problemas, procedimentos de instalação) que são necessárias para lançamento da versão.
- Na conclusão da fase de transição, o incremento torna-se uma versão do software utilizável.

Fases do Processo Unificado



Produção

- Durante essa fase
  - Monitora-se o uso contínuo do software
  - Disponibiliza-se suporte para o ambiente operacional
  - Realiza-se e avalia-se relatórios de defeitos e solicitações de mudanças.

❖ Nesse modelo, é provável que

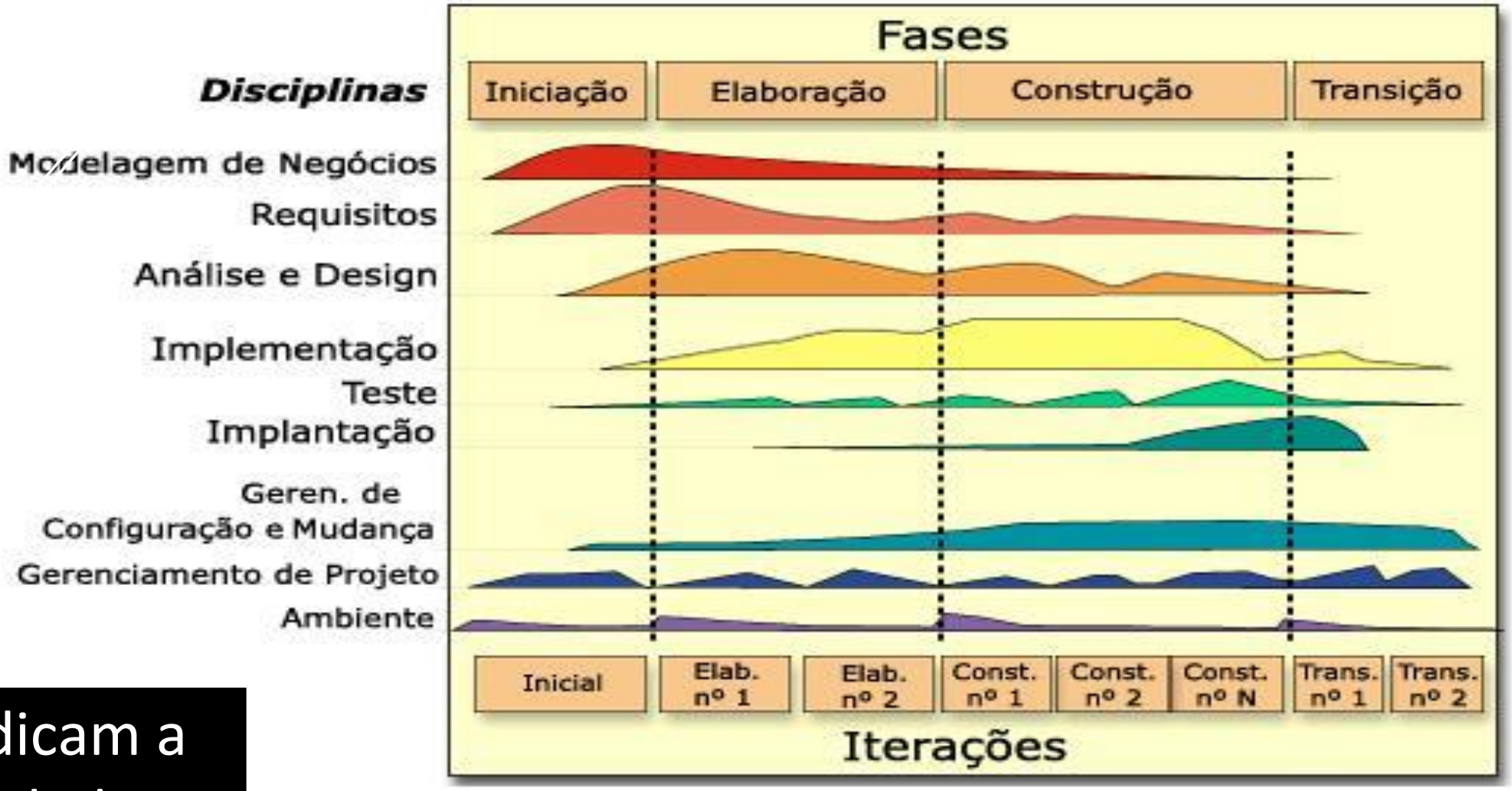
- Ao mesmo tempo em que as fases de construção, transição e produção estejam sendo conduzidas.
  - -Já se tenha iniciado o incremento de software seguinte.
- Isso significa que as cinco fases do PU não ocorrem em sequência, mas sim concorrentemente.

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

Para desenvolver em

Pode ser em grupo.

- **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

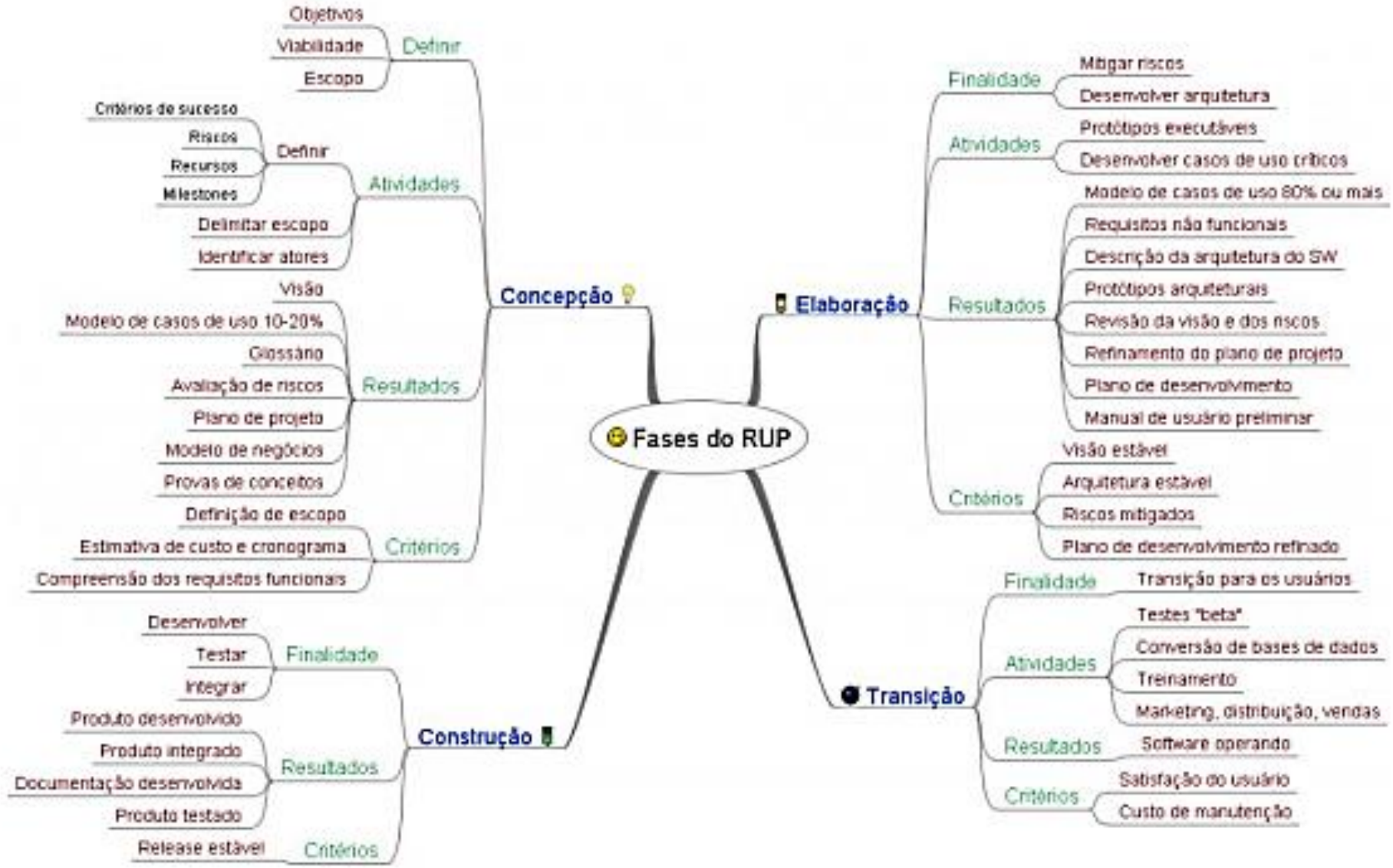
Fluxos de Trabalho (Workflows)

- **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.**

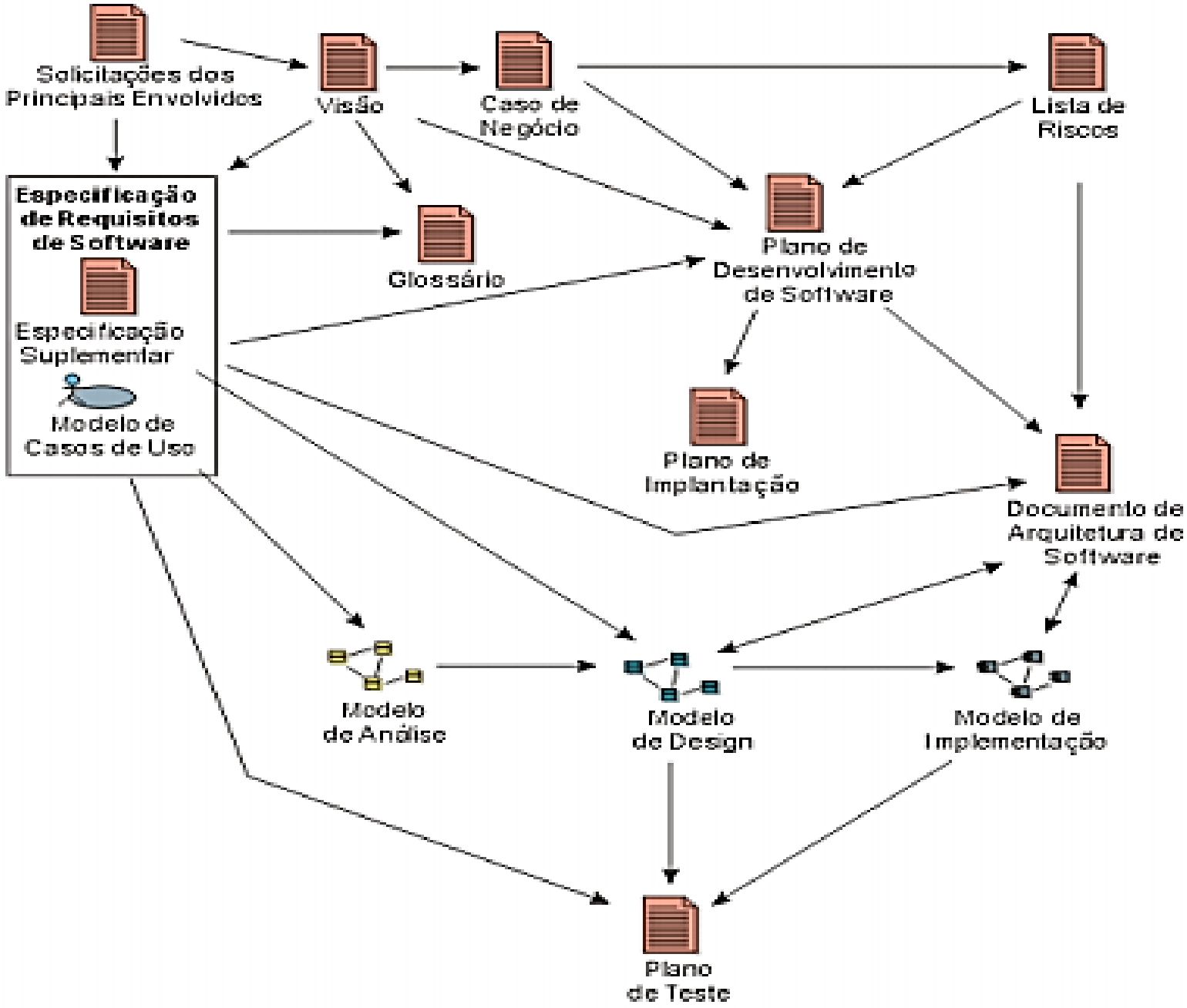
✓**Planos de iteração, que mostram como a iteração deverá ser executada.**



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.

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?**

FMU
CENTRO UNIVERSITÁRIO

Metodologias Ágeis - Introdução

Agil

O Manifesto Ágil

- Ano de 2001
  - Kent Beck e outros dezesseis renomados desenvolvedores, autores e consultores da área de software - batizados de “Agile Alliance”- “Aliança dos Ágeis” assinaram o “Manifesto para o Desenvolvimento Ágil de Software”, que se inicia da seguinte maneira:

❑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

O processo de Software baseado em Metodologias Ágeis

- Foram desenvolvidos para sanar fraquezas reais e perceptíveis da engenharia de software convencional.

- O desenvolvimento ágil oferece benefícios importantes, no entanto, não é indicado para todos os projetos, produtos, pessoas e situações.

FMU
CENTRO UNIVERSITÁRIO

O processo de Software baseado em Metodologias Ágeis

- As condições de mercado mudam rapidamente, as necessidades dos usuários finais se alteram e novas ameaças competitivas emergem sem aviso.

- Em muitas situações, não se conseguirá definir completamente requisitos antes que se inicie o projeto.

- É preciso ser ágil o suficiente para dar uma resposta ao ambiente de fluido negócios.

O processo de Software baseado em Metodologias Ágeis

- Fluidez implica mudanças, e mudanças são caras. Particularmente, se forem sem controle e mal gerenciadas.

- Uma das características mais convincentes da abordagem ágil é sua habilidade de reduzir os custos da mudança ao longo de todo o processo de software.

O processo de Software baseado em Metodologias Ágeis

Vamos então esquecer os princípios, conceitos, métodos, e ferramentas da Engenharia de Software para aderir o processo de desenvolvimento e manutenção de software para nos adequar aos desafios de uma organização político-econômica, globalizada, de extremo dinamismo???

Pressman responde: ABSOUTAMENTE NÃO!!!

O processo de Software baseado em Metodologias Ágeis

A proposta do modelo ágil é para fazer frente aos modelos tradicionais, mais prescritivos, que possuem certa rigidez.

A engenharia de software ágil **combina <u>filosofia</u> com um conjunto de <u>princípios de desenvolvimento</u>**.

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

Etapas Envolvidas no Método Ágil

As atividades metodológicas básicas permanecem:

- **Comunicação,**
- **Planejamento,**
- **Modelagem,**
- **Construção e**
- **Emprego.**

Entretanto, estas se transformam em um conjunto de **tarefas mínimas** que **impulsiona** a equipe para o desenvolvimento e para a **entrega**

Agilidade

- Palavra da Moda para se descrever um moderno processo de software.
- Uma equipe ágil é aquela rápida e capaz de responder apropriadamente a mudanças.
  - Mudanças têm muito a ver com desenvolvimento de software.
- Mudanças
  - → no software que está sendo criado;
  - → nos membros da equipe;
  - → com novas tecnologias;
  - → de todos os tipos que geram impactos no produto em construção ou no projeto.
- Suporte p/ mudanças deve ser incorporado em tudo o que fazemos em software, algo que abraçamos porque é o coração e a alma do software.
- Uma equipe ágil reconhece que o software é desenvolvido por indivíduos trabalhando em equipes e que as habilidades dessas pessoas, suas capacidades em colaborar estão no cerne do sucesso do projeto.

Agilidade

- Consiste em algo mais que uma resposta à mudança
- Abrangendo a filosofia no manifesto
  - ✓ Ela incentiva a estruturação e as atitudes em equipe → comunicação mais fácil — entre membros da equipe, entre o pessoal ligado à tecnologia e o pessoal da área comercial, entre os engenheiros de software e seus gerentes.
  - ✓ Enfatiza a entrega rápida do software operacional.
  - ✓ Assume o cliente como parte da equipe de desenvolvimento .
  - ✓ Trabalha para eliminar a atitude de “nós e eles”
  - ✓ Reconhece que o planejamento em um mundo incerto tem seus limites e que o plano (roteiro) de projeto deve ser flexível.

O Processo Ágil

- Como criar um processo capaz de administrar a imprevisibilidade?

- Segundo a filosofia ágil, a resposta está na adaptabilidade do processo.

- Portanto o processo ágil deve ser adaptável!

- Deve adaptar incrementalmente:

  - A equipe ágil precisa de feedback do cliente (de modo que as adaptações apropriadas possam ser feitas).
  - Um efetivo catalisador para feedback de cliente pode ser um protótipo operacional ou parte de um sistema operacional.

O Processo Ágil

- **Deve se instituir uma *estratégia de desenvolvimento incremental.***

- ***Os incrementos de software (protótipos executáveis ou partes* de um sistema operacional) devem ser entregues em curtos períodos de tempo, de modo que as adaptações acompanhem o mesmo ritmo das mudanças.**

- **O cliente consegue avaliar o incremento de software regularmente, fornecer o feedback necessário para a equipe de software e influenciar as adaptações de processo feitas para incluir adequadamente o feedback.**

AGILE MINDSET

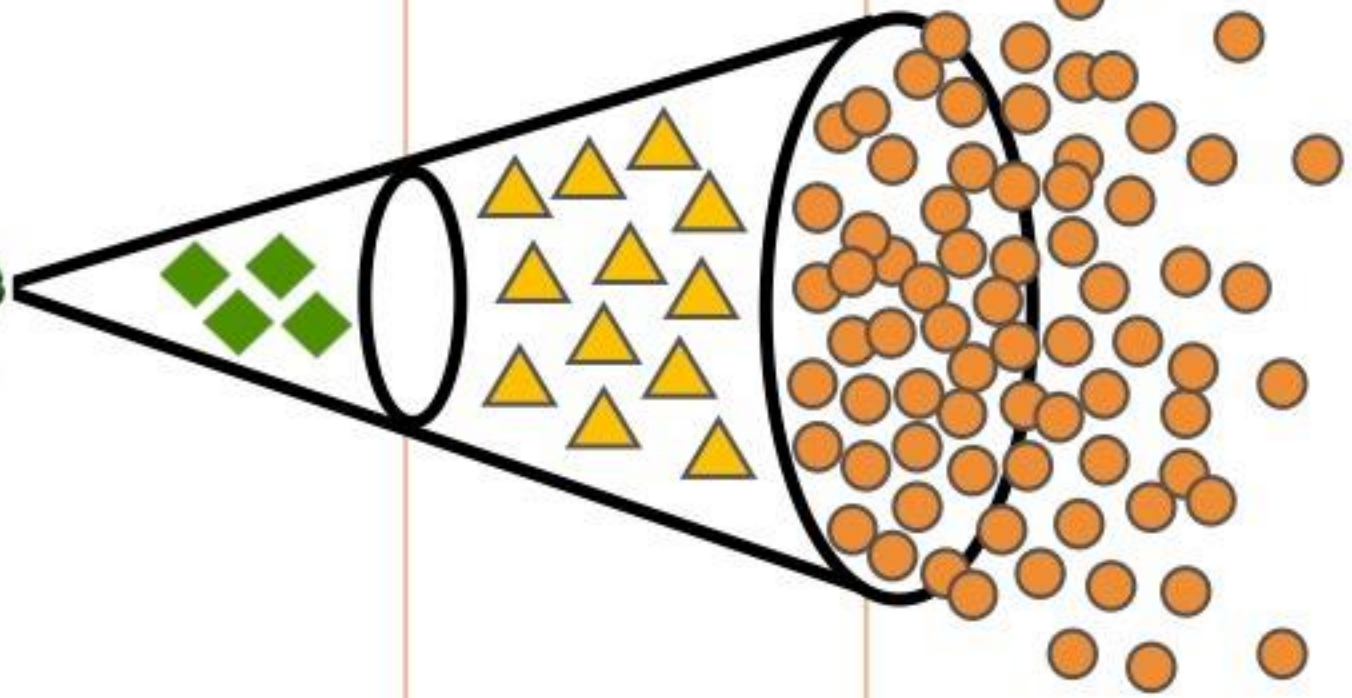
agile é um **MINDSET**

estabelecido por **4 VALORES**

fundamentados por **12 PRINCÍPIOS**

Manifestada através de diversas práticas diferentes

SER ÁGIL



FAZER ÁGIL

mindset | valores | princípios | práticas

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

Manifesto Ágil: Princípios

1. Nossa maior prioridade é satisfazer o cliente através da entrega rápida e contínua de um software de valor
2. Pessoas de negócio e programadores devem trabalhar juntos, diariamente, ao longo de todo o projeto
3. Aceite as mudanças de requisitos, mesmo que numa etapa avançada do desenvolvimento
4. Entregue novas versões do software frequentemente
5. O software em funcionamento é a medida primária de progresso do projeto
6. Construa projetos com pessoas motivadas. Ofereça a elas o ambiente e todo o apoio necessários e acredite em sua capacidade de realização do trabalho

FMU
CENTRO UNIVERSITÁRIO

Manifesto Ágil: Princípios

7. As melhores arquiteturas, requisitos e projetos emergem de equipes auto-organizadas
8. O método mais eficiente e efetivo de distribuir a informação para e entre uma equipe de desenvolvimento é a comunicação face a face
9. Processos ágeis promovem desenvolvimento sustentado
10. A atenção contínua na excelência técnica e num bom projeto aprimora a agilidade
11. Simplicidade é essencial
12. Equipes de projeto avaliam seu desempenho em intervalos regulares e ajustam seu comportamento de acordo com os resultados