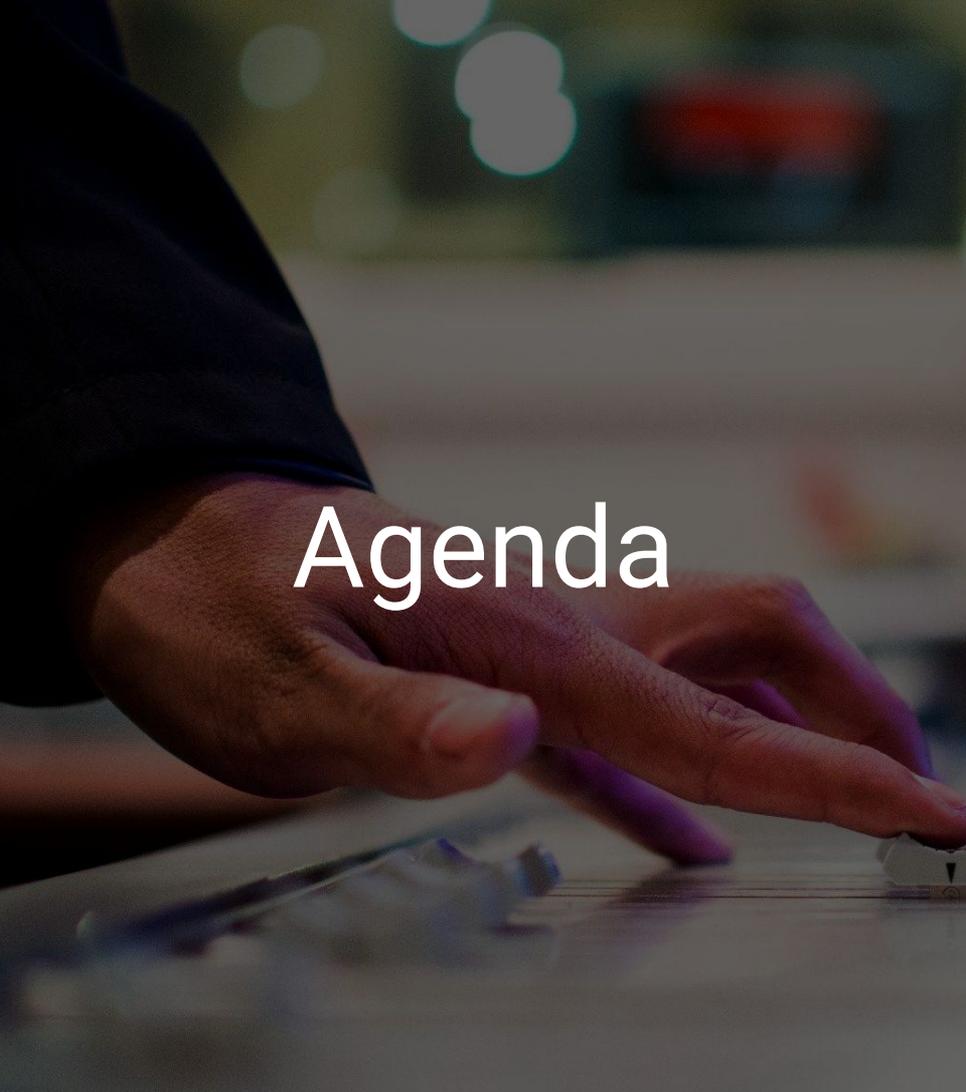


Processos e Modelos e Software

Prof: Nilson Júnior

nilson.junior@jaboatao.ifpe.edu.br

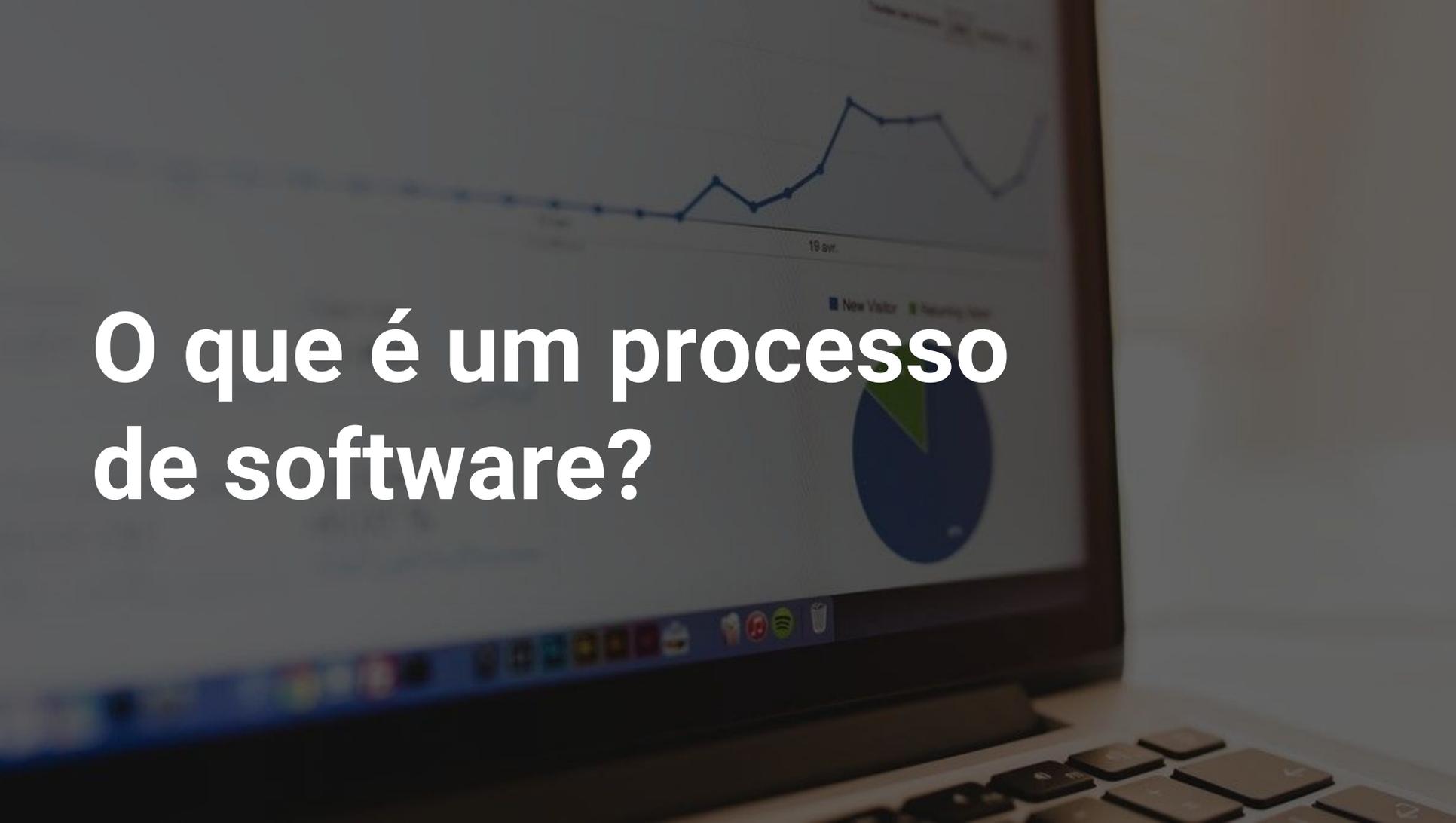
www.professornilson.wix.com/ifpe

A close-up photograph of a person's hands writing on a whiteboard with a marker. The background is blurred, showing some bokeh lights. The word 'Agenda' is overlaid in white text on the left side of the image.

Agenda

- Processo de Software em **Cascata**.
- Processo de Software **Iterativo e Incremental**
- Processo de Software baseado em **componente**
- Processo de Software em **Espiral**
- Processo de Software em **Ágeis**
- Apresentação do **RUP**

O que é um processo de software?



Processo de Software

É um conjunto estruturado de atividades, procedimentos, artefatos e ferramentas necessários para o desenvolvimento de um sistema de software.

Atividades Genéricas: Especificação, Projeto, Validação, Desenvolvimento e Evolução

Modelos genéricos de processo de software

Modelo Cascata

Fases separadas e distintas de especificação e desenvolvimento.

Desenvolvimento iterativo

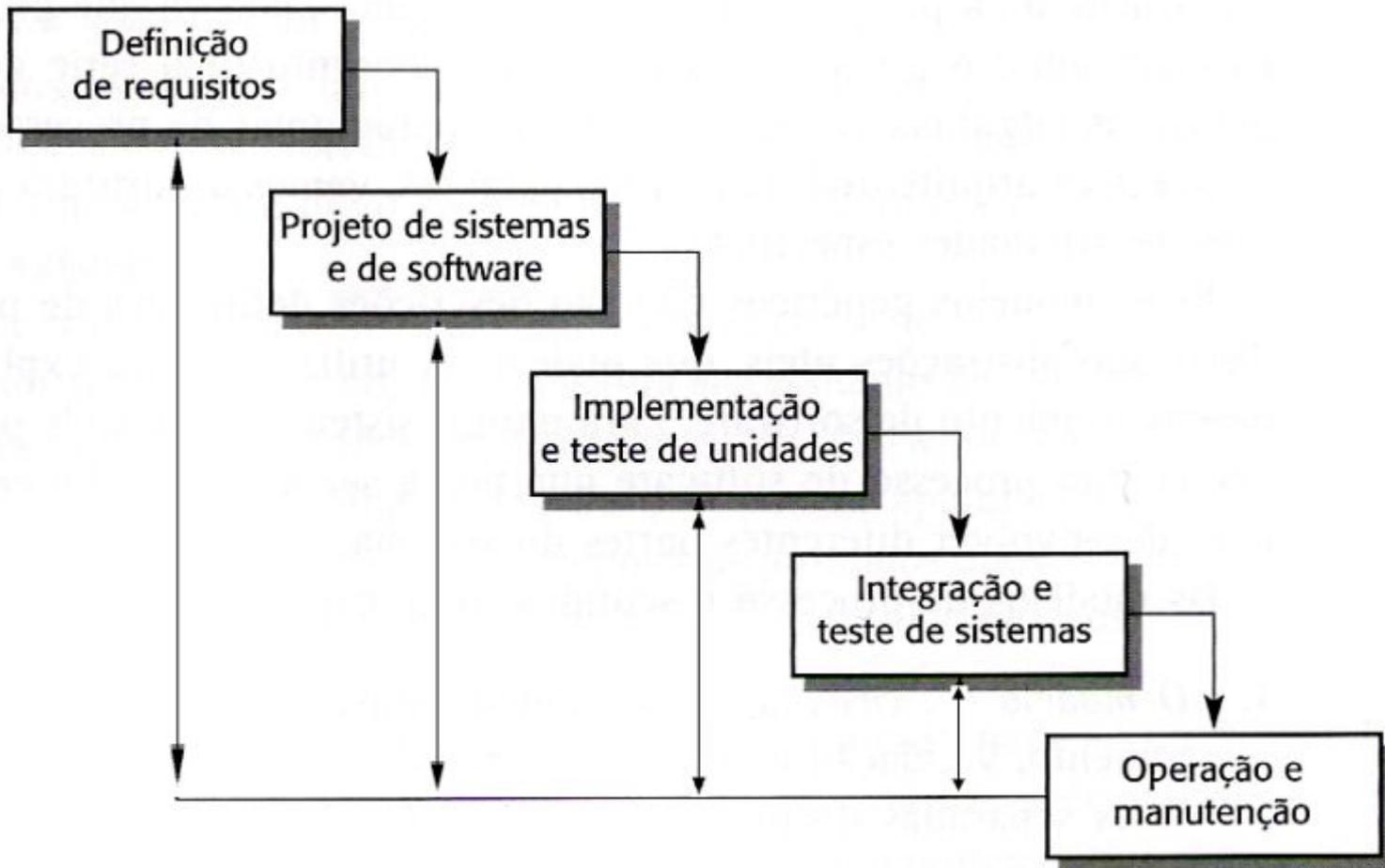
Sistema desenvolvido através de várias etapas

Engenharia de software baseada em componentes

O sistema é montado a partir de componentes existentes.

A background image of a waterfall cascading down a rocky ledge into a pool of water, surrounded by lush green foliage. The image is overlaid with a semi-transparent green filter.

Modelo Cascata



Fases do Modelo Cascata

- Análise e definição de requisitos
 - Projeto de sistema e software
 - Implementação e teste de unidade
 - Integração e teste de sistema
 - Operação e manutenção
- Primeiro modelo a organizar as atividades de desenv.
 - Uma fase tem de estar completa antes de passar para a próxima.

Particionamento **inflexível** do projeto em estágios

Documentos “**completamente elaborados**” são necessários para fazer as transições entre estágios
Apropriado somente quando os requisitos são bem compreendidos e quando as mudanças são raras

O modelo cascata é o **mais usado em projetos de engenharia de sistemas de grande porte**, onde um sistema é desenvolvido em várias localidades.

Engenharia de Software baseada em Componentes

A 3D illustration of a person sitting at a desk with a large gear, symbolizing software engineering or component-based design. The person is rendered in a simple, rounded style, and the gear is a large, dark brown object. The background is a dark green color with a subtle pattern of faint, light green lines and shapes, suggesting a technical or digital environment.

Modelo baseado em componentes

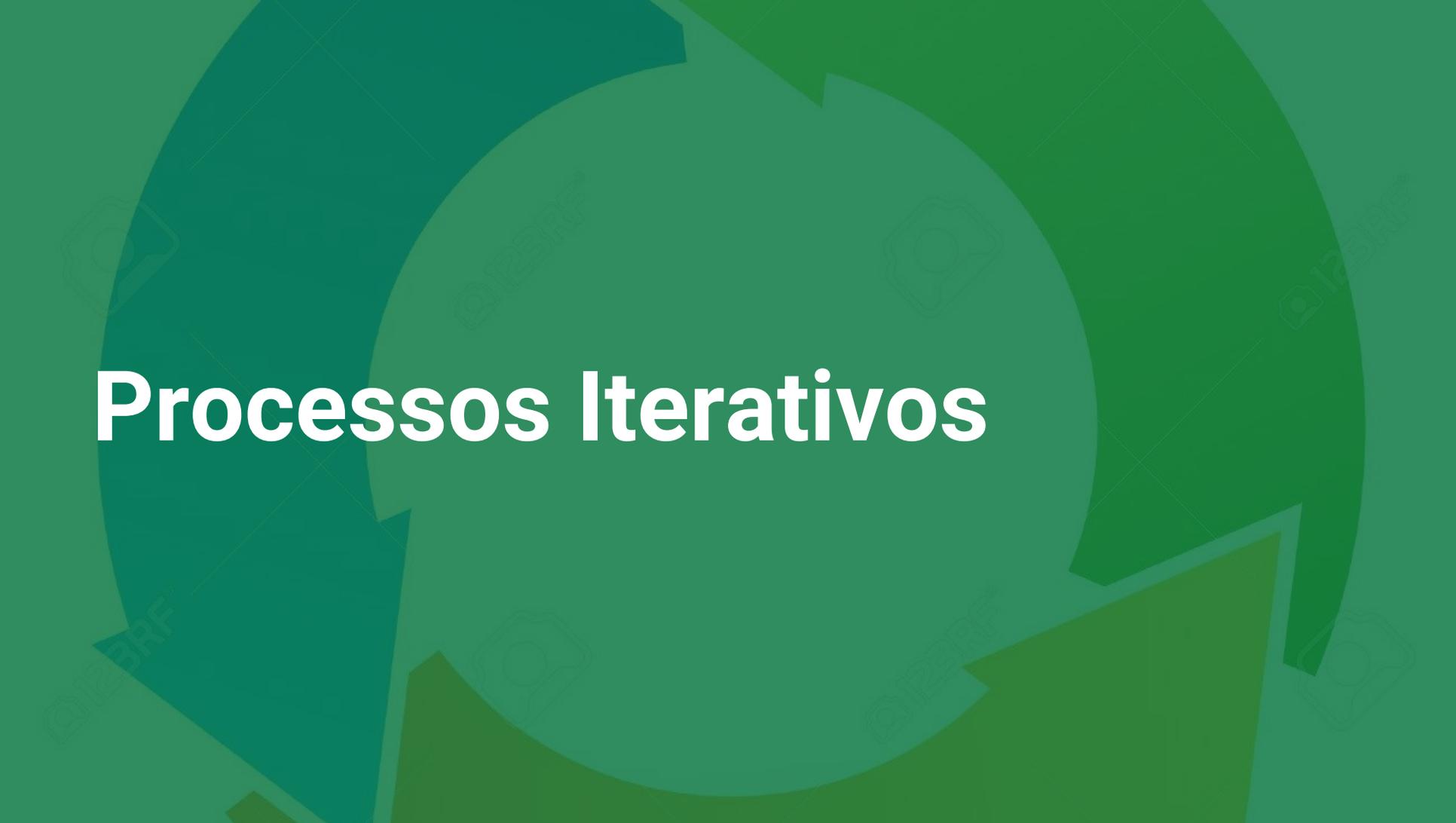


Baseado em reuso sistemático onde sistemas são integrados a partir de componentes existentes ou de sistemas COTS (Commercial-of-the-shelf)

Estágios do modelo

- Análise de componentes;
- Modificação de requisitos;
- Projeto de sistema com reuso;
- Desenvolvimento e integração;

Esta abordagem está se tornando cada vez mais usada à medida que padrões de componentes têm surgido.

The background features a large, stylized circular arrow pointing clockwise, composed of two overlapping curved segments in shades of teal and green. In the center of this arrow is a solid light green circle. The overall background is a dark teal color with a subtle grid pattern.

Processos Iterativos

Processo Iterativo

- Requisitos de sistema SEMPRE evoluem no curso de um projeto
- Algum retrabalho é necessário
- A abordagem iterativa pode ser aplicada a qualquer um dos modelos genéricos do processo.

Duas abordagens (relacionadas)

- Entrega incremental;
- Desenvolvimento espiral.

Entregas Incremental

O sistema é entregue ao cliente em incrementos

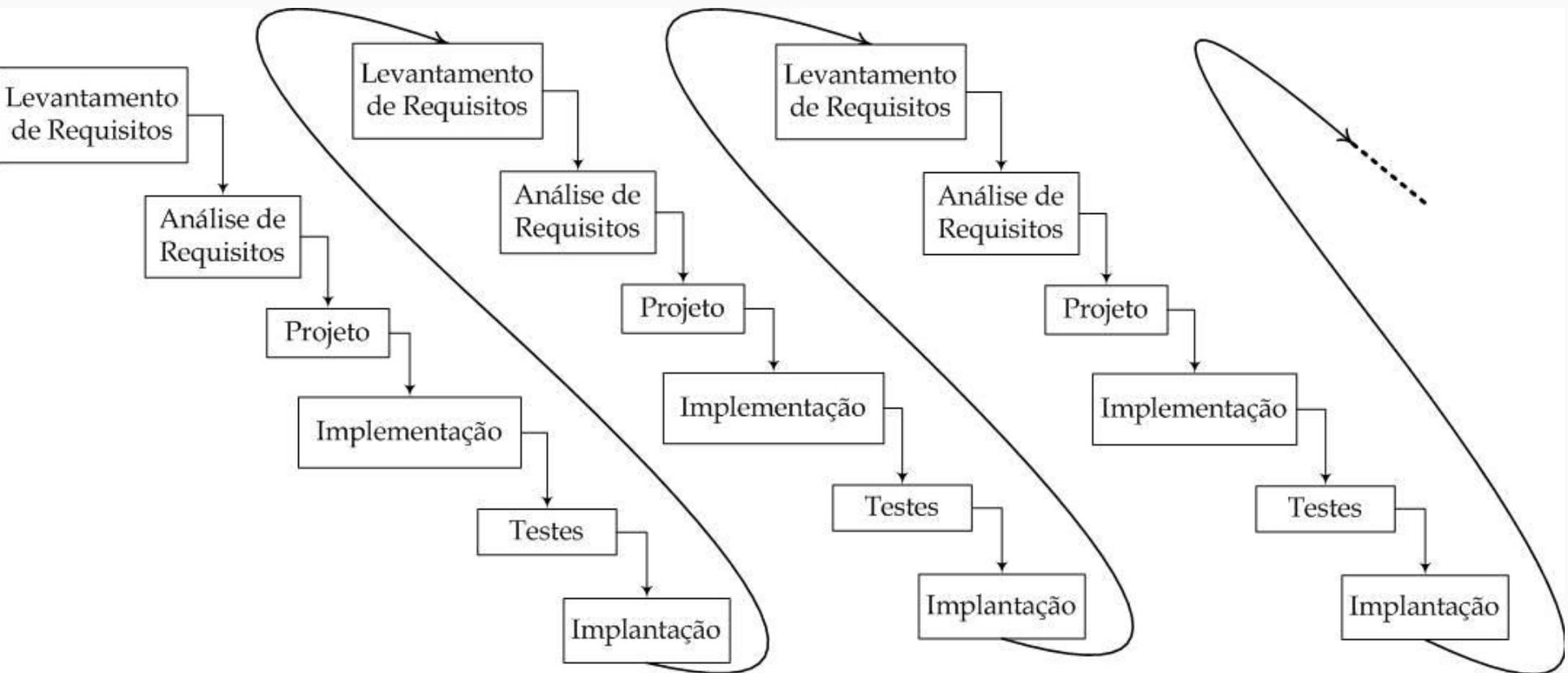
Cada incremento fornece parte da funcionalidade

Os requisitos são priorizados

Requisitos de prioridade mais alta são incluídos nos incrementos iniciais.

Uma vez que o desenvolvimento de um incremento é iniciado, os requisitos são congelados

Os requisitos para os incrementos posteriores podem continuar evoluindo (e incluir requisitos já implementados!)



Vantagens do desenvolvimento Incremental

- Incrementos podem ser entregues regularmente ao cliente e, desse modo, a funcionalidade de sistema é disponibilizada mais cedo.
- Os incrementos iniciais agem como protótipos para elicitare os requisitos para incrementos posteriores do sistema.
- Riscos menores de falha geral do projeto.
- Os serviços de sistema de mais alta prioridade tendem a receber mais testes.

A photograph of two men in an office environment. The man on the left is wearing glasses and a light-colored shirt, looking towards the right. The man on the right is also wearing glasses and a light-colored shirt, pointing his right hand towards a laptop screen. The background shows a wall with several sticky notes and a computer monitor. The entire image is overlaid with a semi-transparent green filter.

Extreme Programming

Extreme Programming

- Uma abordagem baseada no desenvolvimento e na entrega de incrementos de funcionalidade muito pequenos.
- Baseia-se no aprimoramento constante do código, em testes automatizados, no envolvimento do usuário na equipe e no desenvolvimento em pares.
- Processo de desenvolvimento que usaremos ao longo da disciplina

Modelo Espiral

Desenvolvimento Espiral

- O processo é representado como uma espiral ao invés de uma sequência de atividades com realimentação.
- Cada loop na espiral representa uma fase no processo.
- Sem fases definidas, tais como especificação ou projeto – os loops na espiral são escolhidos dependendo do que é requisitado.
- Os riscos são explicitamente avaliados e resolvidos ao longo do processo.

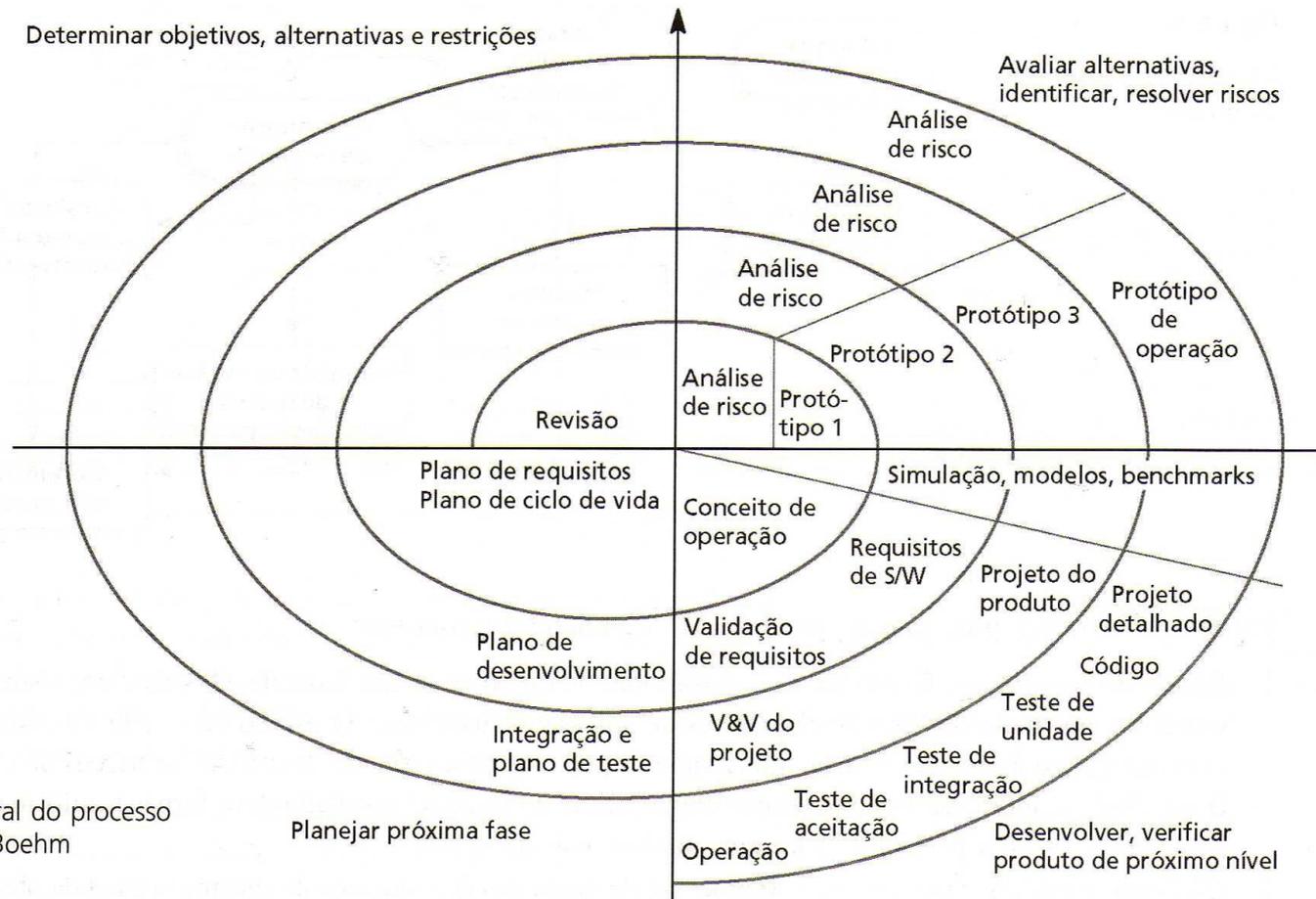


Figura 4.5

Modelo em espiral do processo de software de Boehm (©IEEE, 1988).

Desenvolvimento Espiral

Definição de objetivos

Objetivos específicos para a fase são identificados.

Avaliação e redução de riscos

Riscos são avaliados e atividades são realizadas para reduzir os riscos-chave

Desenvolvimento e validação

Um modelo de desenvolvimento para o sistema, que pode ser qualquer um dos modelos genéricos, é escolhido.

Planejamento

O projeto é revisado e a próxima fase da espiral é planejada.



Atividades Genéricas de um Processo de Desenvolvimento

Atividades de um Processo de Software

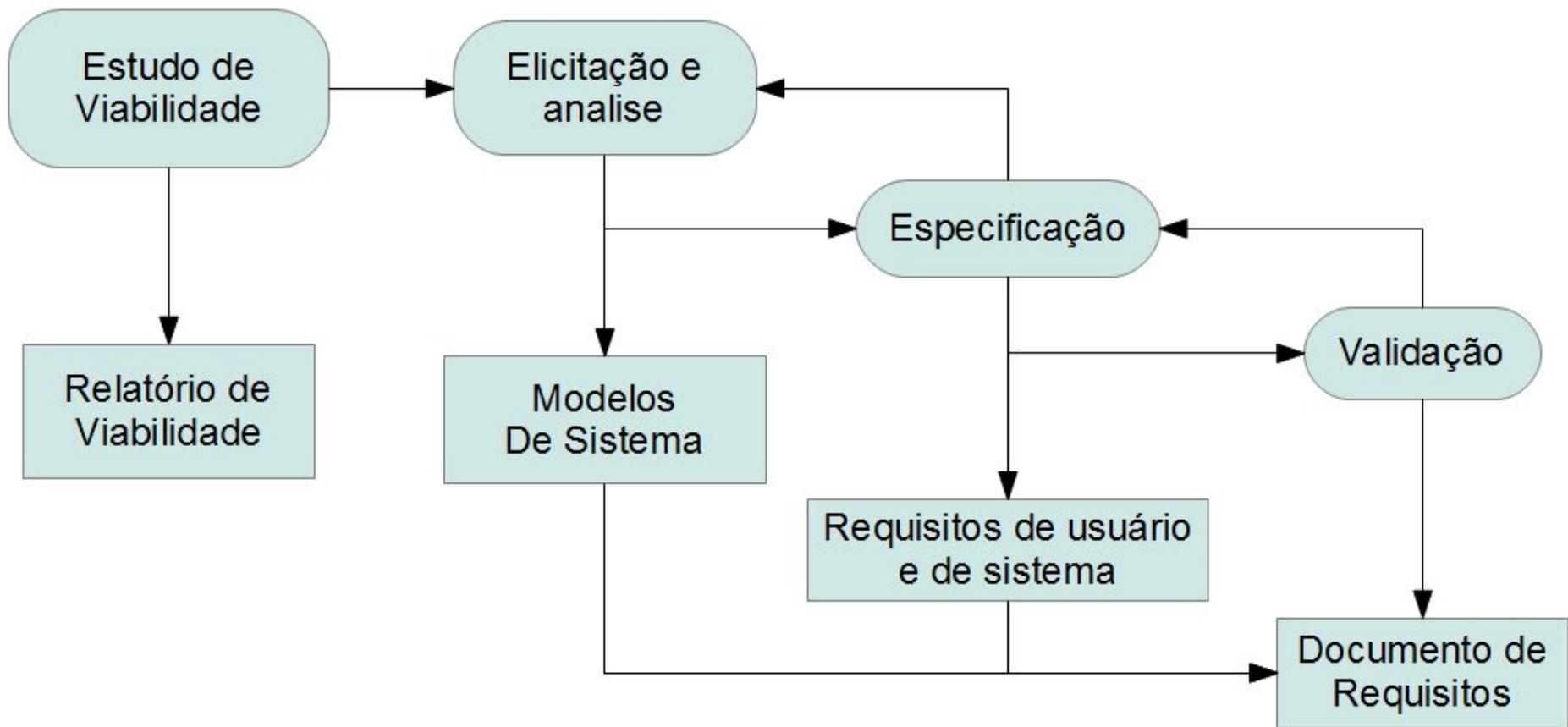
- Especificação de software
- Projeto e implementação de software
- Validação de software
- Evolução de software

Especificação de Software

É o processo para definir quais serviços são necessários e identificar as restrições de operação e de desenvolvimento do sistema.

Processo de engenharia de requisitos

- Estudo de viabilidade realizado antes do projeto ser iniciado.
- Elicitação e análise de requisitos;
- Especificação de requisitos;
- Validação de requisitos.



Atividades de projeto

- *Projeto de arquitetura*
- *Projeto de interfaces*
- *Projeto de componente*
- *Projeto de dados*
- *Projeto de algoritmo*

Projeto e Implementação de Software

É o processo de conversão da especificação em um sistema de software

Projeto de software

Projetar uma estrutura de software que atenda à especificação.

Implementação

Transformar essa estrutura em um programa executável

As atividades de projeto e implementação são fortemente relacionadas e podem ser intecaladas.

Programação e Depuração

1. É a transformação de um projeto em um programa e a remoção de defeitos desse programa.
2. Programação é uma atividade pessoal – não há processo genérico de programação.
3. Há algumas práticas, porém, que são universalmente consideradas boas
4. Programadores realizam alguns testes para descobrir defeitos no programa e removem esses defeitos no processo de depuração

Estágios de teste

Teste de componente ou unidade

Os componentes individuais são testados independentemente;
Esses componentes podem ser funções ou classes de objetos, ou grupos coerentes dessas entidades.

Teste de sistemas

Teste de sistema como um todo. O teste das propriedades emergentes é particularmente importante.

Teste de Aceitação

Teste com dados do cliente para verificar se o sistema atende às suas necessidades.

Validação de software

Verificação e validação (V & V) têm a intenção de mostrar que um sistema está em conformidade com a sua especificação e que atende aos requisitos do cliente

Verificação: “construímos o sistema corretamente?”

Exs: inspeção de código, análise estática

Validação: “construímos o sistema correto?”

Exs: testes, animação de especificações

Testes envolvem a execução do sistema com casos de teste que são derivados da especificação do sistema e de dados reais a ser processados por ele.

O software é inerentemente flexível e pode mudar.

Requisitos mudam devido a diversos fatores e o software deve acompanhar essas mudanças.

Processos antigos separavam explicitamente desenvolvimento de evolução

- Processos e métodos iterativos (XP, RUP, Espiral) normalmente não fazem uma separação explícita

Evolução pode se dever a diversas razões:

- Correções (patches) Mudanças de requisitos
- Melhoria de funcionalidades pré-existentes

DISCIPLINES

BUSINESS MODELLING

REQUIREMENTS

ANALYSIS & DESIGN

IMPLEMENTATION

TEST

DEPLOYMENT

CONFIGURATION & CHANGE MANAGEMENT

PROJECT MANAGEMENT

ENVIRONMENT

Inception

Elaboration

Construction

Transition

RUP - Rational Unified Process

Initial

Elab #1

Elab #2

Const #1

Const #2

Const #3

Tran #1

Tran #2

ITERATIONS

Rational Unified Process

É um modelo de processo moderno baseado na UML

Tenta cobrir todos os aspectos do desenvolvimento de software

Fortemente focado na documentação do sistema

Normalmente descrito a partir de três perspectivas:

- Uma perspectiva dinâmica que mostra as fases ao longo do tempo;
- Uma perspectiva estática que mostra atividades de processo;
- Uma perspectiva prática que sugere bons princípios e práticas de desenvolvimento

Concepção

Estabelecer o escopo e viabilidade econômica do projeto



Elaboração

Eliminar principais riscos e definir arquitetura estável



Construção

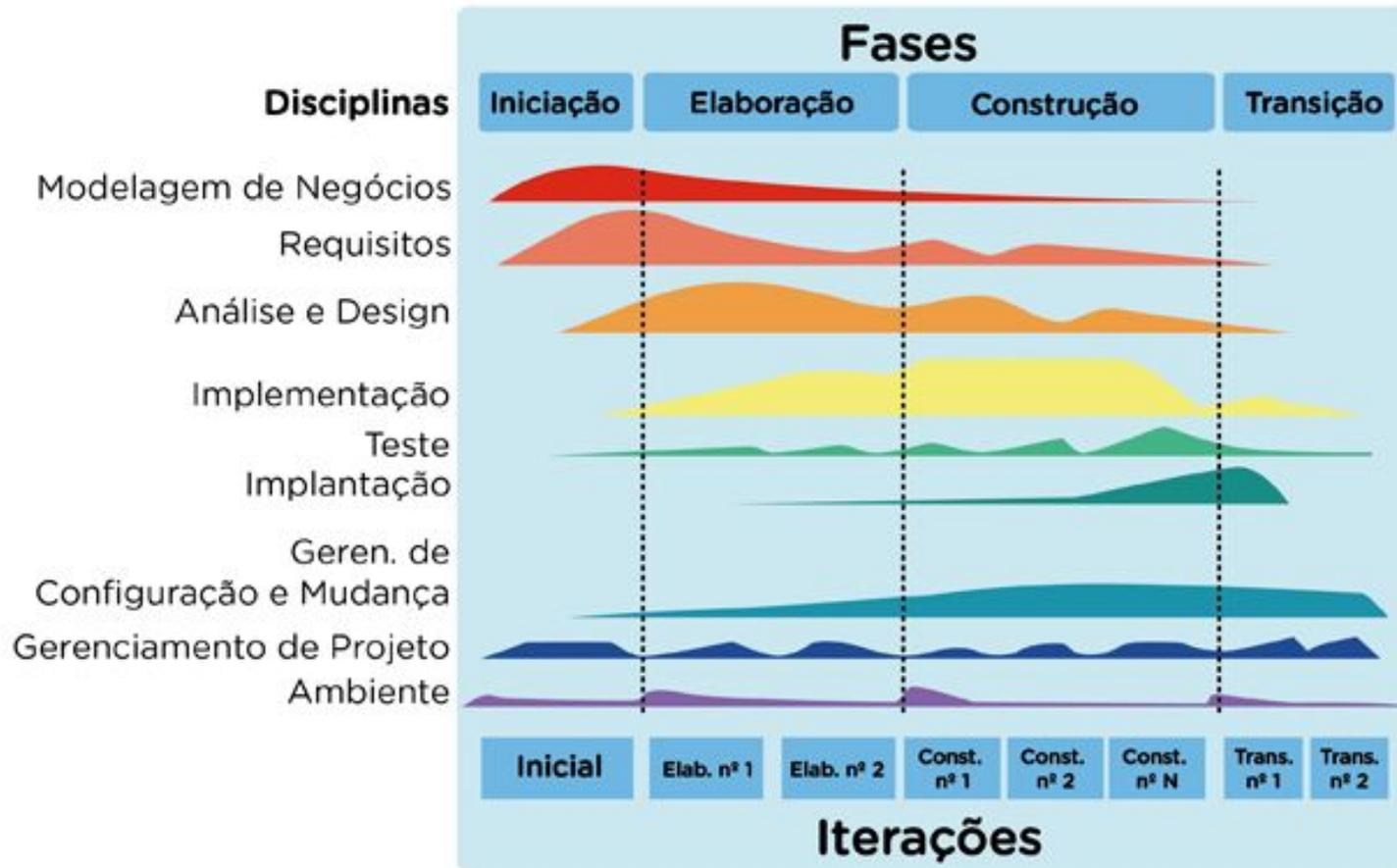
Desenvolver o produto até que ele esteja pronto para beta testes



Transição

Entrar no ambiente do usuário





Boas práticas do RUP

- **Desenvolver o software iterativamente**
- **Gerenciar requisitos**
- **Usar arquiteturas baseadas em componentes**
- **Modelar o software visualmente**
- **Verificar a qualidade de software**
- **Controlar as mudanças do software**

OpenUP

Passo 1

Acesse o site:

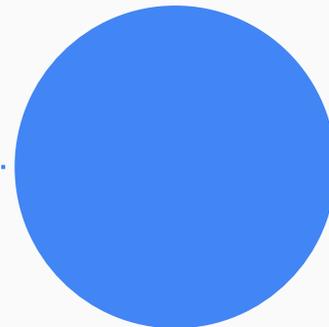
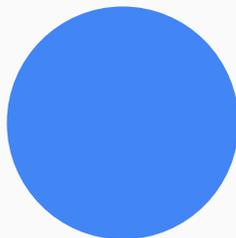
<http://www.wthreeex.com/rup/OpenUP/>

Passo 2

Acesse as fases do RUP

Passo 3

Acesse as Atividades de cada fase



Dúvidas ?

