



Processo de Desenvolvimento de Software

Profº - Dr. Thales Levi Azevedo Valente
thales.l.a.valente@ufma.com.br

Sejam Bem-vindos !



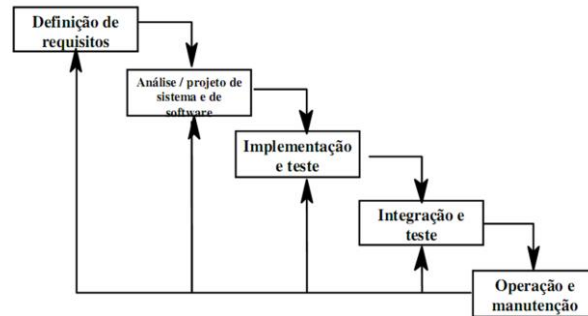
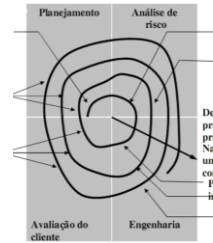
**Os celulares devem
ficar no silencioso
ou desligados**

Pode ser utilizado
apenas em caso
de emergência



**Boa tarde/noite, por
favor e com licença
DEVEM ser usados**

Educação é
essencial



Na aula anterior...



Objetivos de hoje



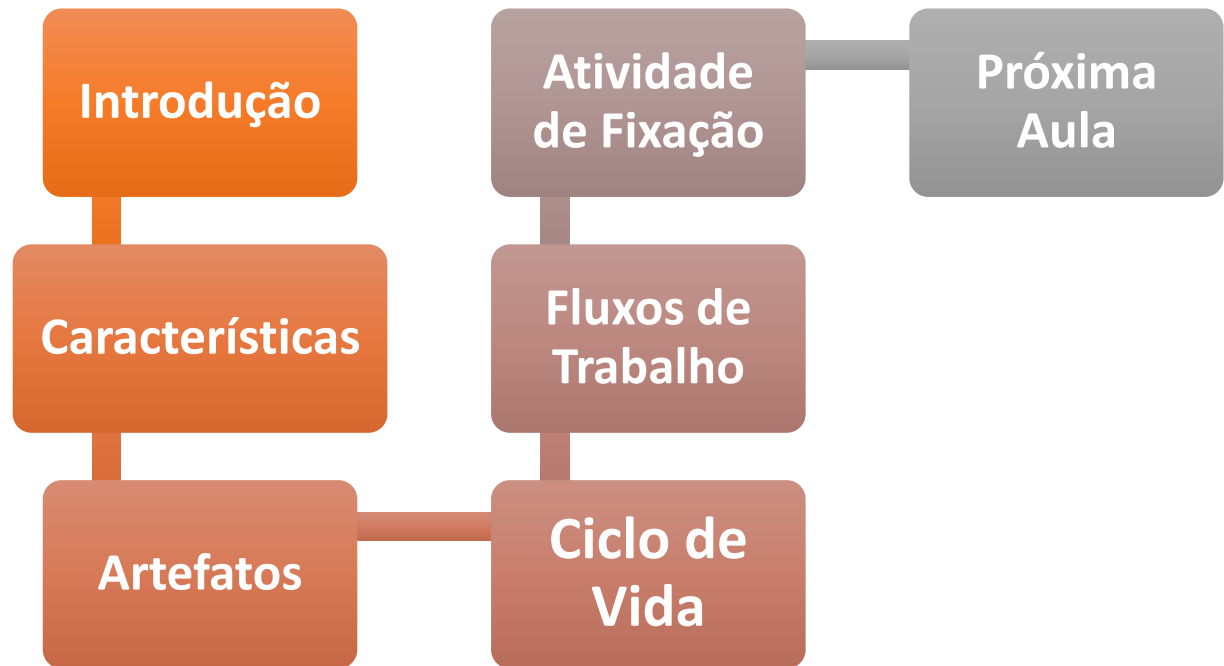
Oferecer uma visão geral do processo unificado



Ao final da aula, os alunos serão capazes de entender o que é o processo unificado, as fases de seu ciclo de vida, fluxos de trabalho e artefatos



Roteiro: Processo Unificado



Introdução ao Processo Unificado (PU)

- PU encaixa-se na definição de processo: um conjunto de atividades executadas para transformar um conjunto de requisitos do cliente em um sistema de software.
- PU também é uma estrutura genérica de processo
 - Pode ser customizado adicionando-se ou removendo-se atividades com base nas necessidades específicas e nos recursos disponíveis para o projeto.

Características do PU

- É um framework genérico de um processo de desenvolvimento
- É baseado em componentes que realizam as interfaces
- Utiliza toda a definição da UML
- Alicerces
 - Dirigido por Casos de Uso
 - Centrado na Arquitetura
 - Desenvolvimento iterativo e incremental
- Os 4 Ps: **peçoal**, **projeto**, **produto** e **processo**.

P4 = Pessoa, Projeto, Produto, Processo

- **PESSOAS** financiam, escolhem, desenvolvem, gerenciam, testam, usam e são beneficiadas por produtos
- **PROJETOS** sofrem alterações. Determinam os tipos de **pessoas** que irão trabalhar no projeto e os artefatos que serão usados



P4 = Pessoa, Projeto, Produto, Processo

- **PRODUTO** código fonte, código de máquina, subsistemas, classes, diagramas: interação, de estados e outros **artefatos**
- **PROCESSO** define **quem** (**papel**) faz **o que** (**artefato**), **quando** (**disciplina**) e **como** (**atividades**)
- **PU** é um processo. Considera fatores organizacionais, do domínio, ciclo de vida e técnicos

Artefatos

- **ARTEFATO** é qualquer tipo de **informação criada por uma pessoa** (diagramas UML, textos, modelos de interfaces)
- Artefatos são quaisquer dos documentos produzidos durante o desenvolvimento, tais como modelos, diagramas, documentos de especificação de requisitos, código fonte ou executável, planos de teste, etc
- Artefatos devem ser controlados e administrados corretamente durante o desenvolvimento do sistema.

Artefatos

- Muitos dos artefatos são opcionais, produzidos de acordo com as necessidades específicas de cada projeto
- Exemplos:
 - Diagrama de casos de uso
 - Diagrama de Classes
 - Diagrama de Sequência
 - Código fonte

Trabalhadores e Atividades

- Trabalhadores

- Um **trabalhador** é alguém que desempenha um papel e é responsável pela realização de atividades para produzir ou modificar um artefato.
Exemplos: analista de sistemas, programador, testador etc.

- Atividades

- tarefa que um trabalhador executa a fim de produzir ou modificar um artefato.

Dirigido por Casos de Uso

- Um caso de uso é uma **seqüência de ações, executadas** por um ou mais **atores**, que **produz** um ou mais **resultados** de valor para um ou mais atores.
- Um **ator** é uma **pessoa** ou outro **sistema**.
- O **PU** é dirigido por **casos de uso**, pois utiliza-os para dirigir todo o trabalho de desenvolvimento, desde a captação inicial e negociação dos requisitos até a aceitação do código (testes)

Dirigido por Casos de Uso

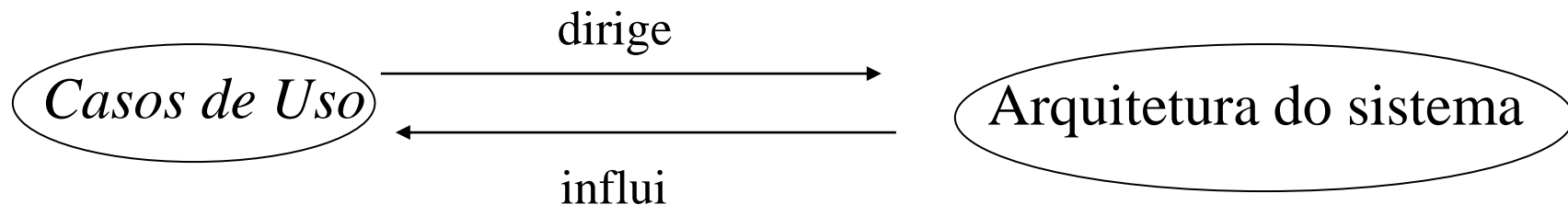
- Os casos de uso possibilitam que os **requisitos funcionais** possam ser **capturados** na **perspectiva de cada um dos usuários** do sistema, e com isso, **definir o comportamento**, as respostas esperadas e os casos de testes que devem validar a **implementação do sistema**.

Dirigido por Casos de Uso

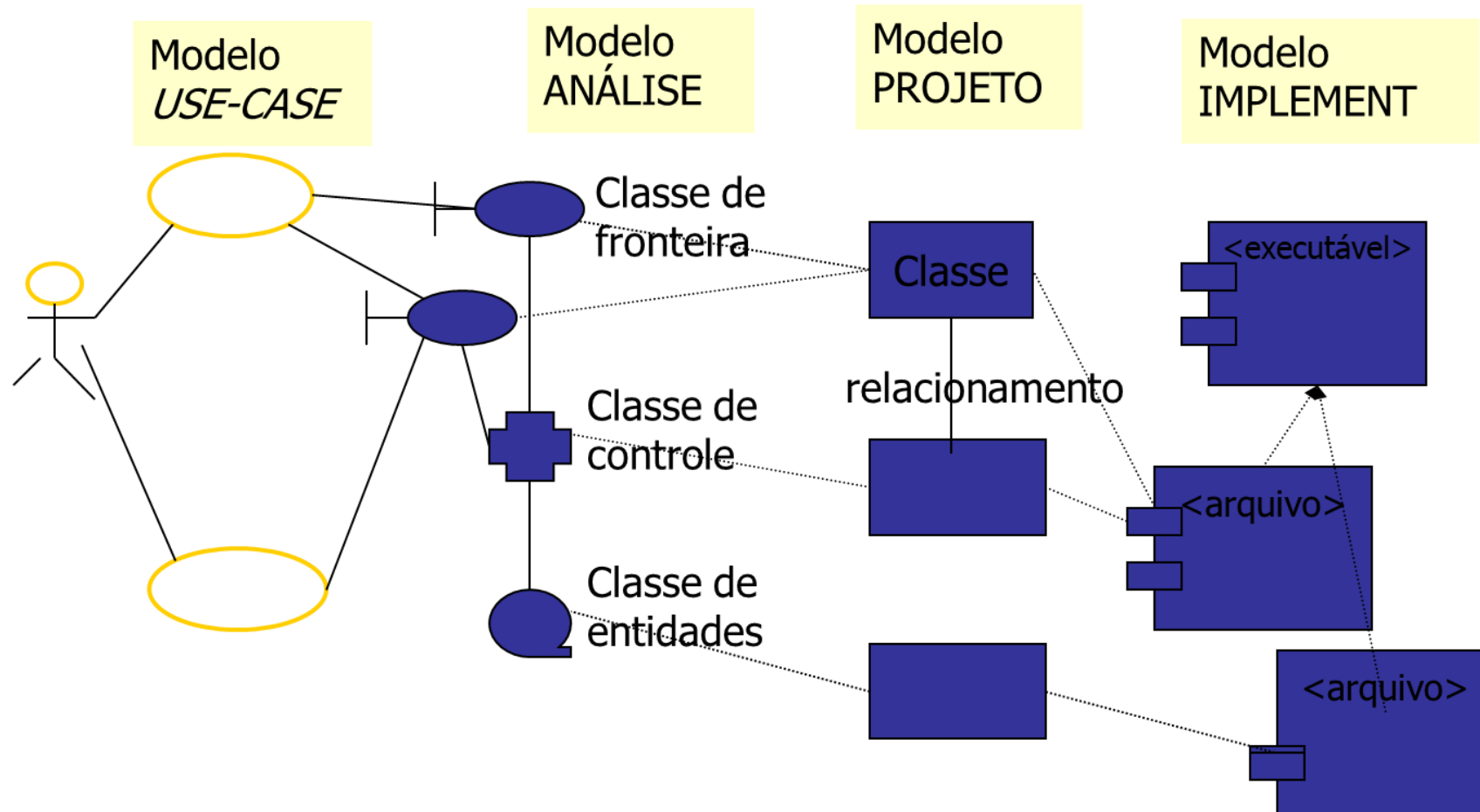
- Os casos de uso acompanham todo o processo de desenvolvimento
 - Especificação de requisitos
 - Análise
 - Projeto
 - implementação e testes.
- Modelo casos de uso = casos de uso = funcionalidade do sistema

Por que Casos de Uso?

- Os requisitos funcionais são registrados preferencialmente por meio deles
- Eles ajudam a **planejar as iterações**
- Eles podem conduzir o projeto
- O teste é baseado neles



Por que Casos de Uso?

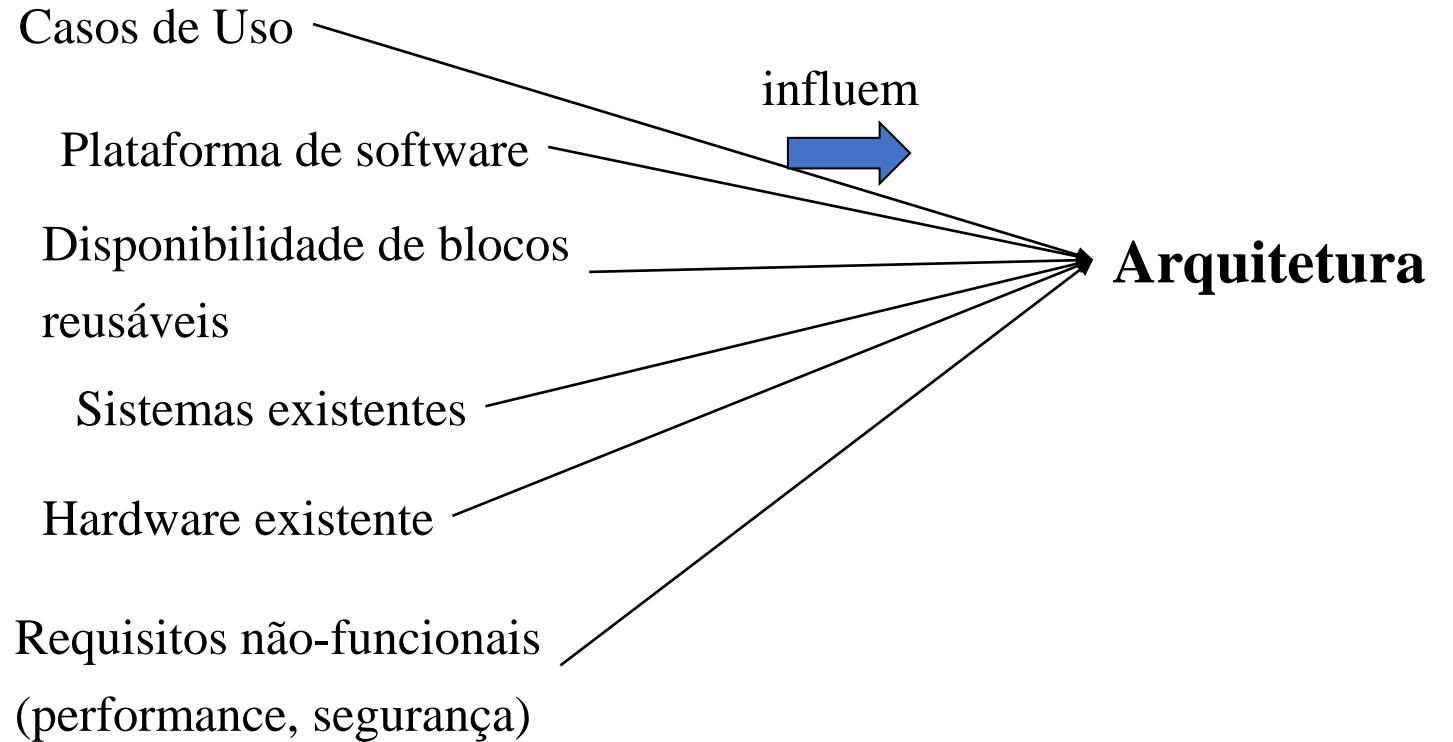


Centrado na arquitetura

- Uma arquitetura é um “**mapa**” do sistema que **define** as diferentes partes do mesmo, seus **relacionamentos**, **interações** e mecanismos de **interconexão**, uma vez que englobam tanto os aspectos estáticos quanto os dinâmicos de um sistema.
- A arquitetura também se refere a questões como **desempenho**, **escalabilidade**, **reuso** e **restrições econômicas e tecnológicas**.
- Por isso, é importante definir uma **arquitetura cedo** e refiná-la com o tempo.

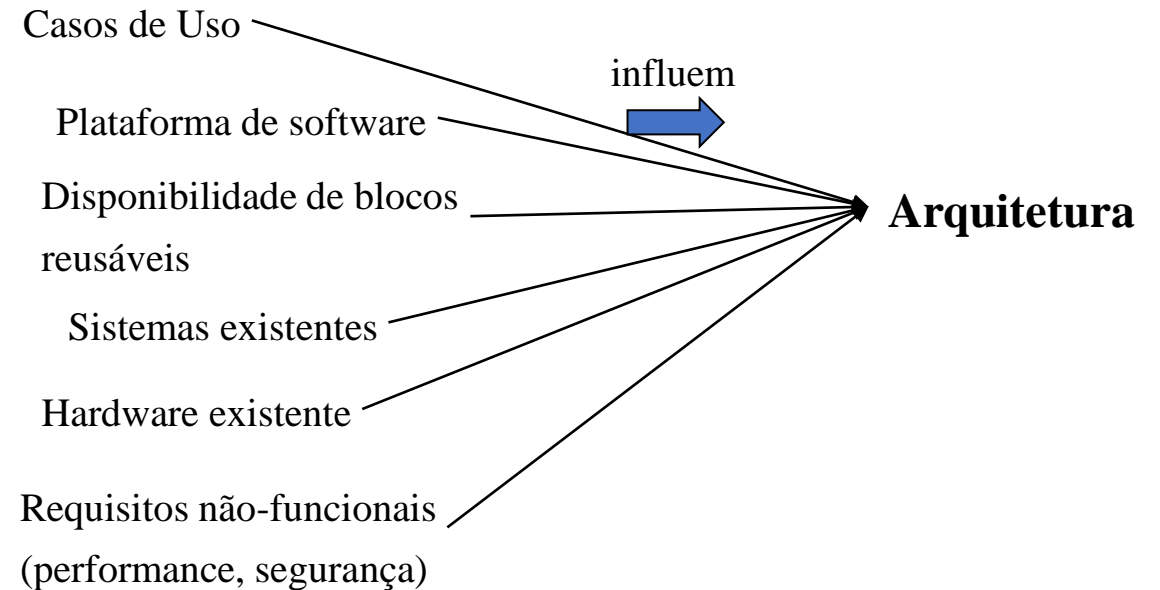
Centrado na arquitetura

- PU é centrado na arquitetura
- É descrita como diferentes visões do sistema
- Casos de uso, sistemas existentes, plataforma de desenvolvimento, framework, etc



Centrado na arquitetura

- A arquitetura é importante
- Ajuda a entender a **visão global**
- Ajuda a **organizar o esforço de desenvolvimento**
- Facilita as **possibilidades de reuso**
- Facilita a **evolução do sistema**
- Tem base nos casos de uso especificados



Desenvolvimento Iterativo e Incremental

- Benefícios
 - Identificação de **riscos** é **adiantada**
 - **Preparação** do sistema para **requisitos** que **mudam**
 - Integração contínua (facilita testes) e aprendizado facilitado
- Resultado de uma iteração: **incremento**

Desenvolvimento Iterativo e Incremental

- O PU é um processo de desenvolvimento **iterativo** e **incremental**, no qual o software **não é implementado** a partir de um instante **no fim do projeto**, mas ao contrário, o ciclo de vida no PU é **desenvolvido e implementado em iterações**.
- Uma **iteração** é um miniprojeto que resulta em uma **versão do sistema liberada** interna ou externamente
 - Essa versão oferece uma melhora incremental sobre a iteração anterior

Desenvolvimento Iterativo e Incremental

- Para cada iteração os desenvolvedores terão que identificar e especificar
 - os casos de usos relevantes
 - desenvolver a atividade usando a arquitetura escolhida como guia
 - implementar o modelo de design em componentes
 - verificar se estes satisfazem os casos de uso.

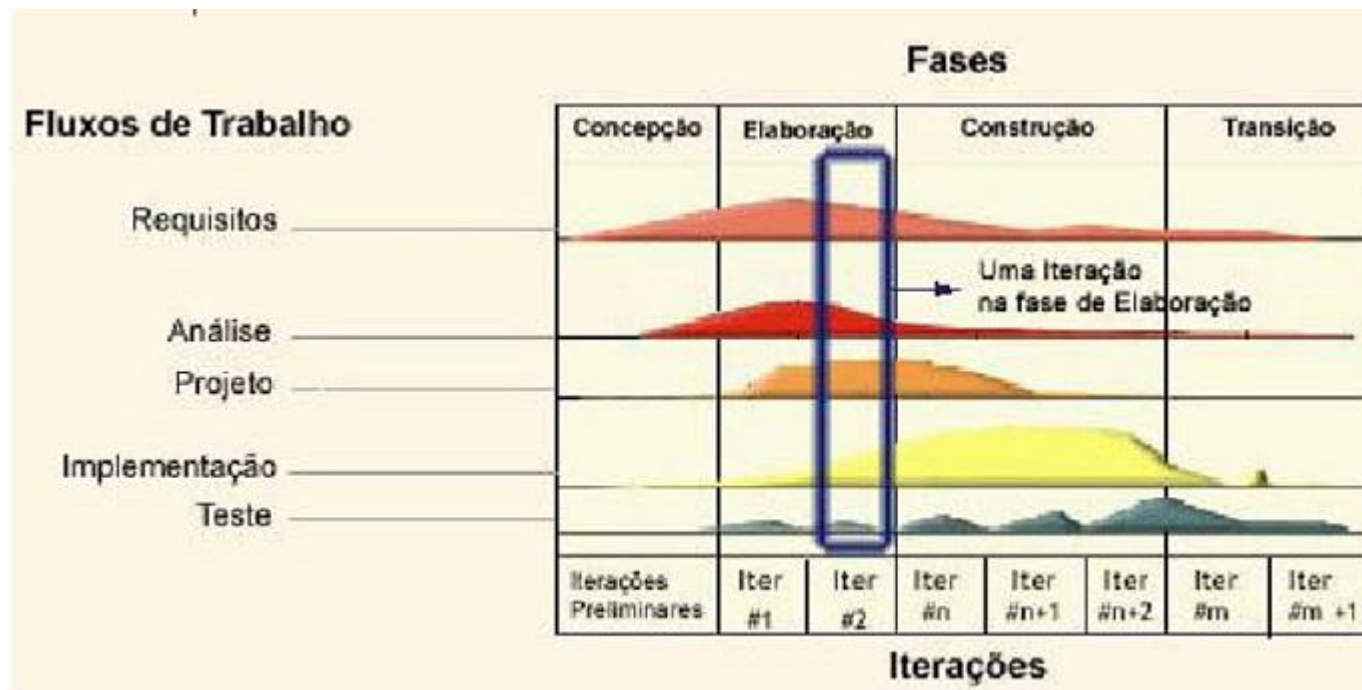
Ciclo de Vida

- PU consiste da repetição de uma série de ciclos durante a vida de um sistema
- Cada ciclo é formado por 4 fases
 - Concepção
 - Elaboração
 - Construção
 - Transição

Ciclo de Vida

- Cada fase, por sua vez, é subdividida em iterações que passam por todos os 5 fluxos do trabalho do processo
 - Requisitos
 - análise
 - Projeto
 - implementação e teste.

Ciclo de Vida



Fase 1 - Concepção

- O objetivo é estabelecer a viabilidade do sistema proposto.
- Nessa fase se faz
 - Definição do escopo do sistema.
 - Estimativas de custos e cronograma
 - Identificação dos potenciais riscos que devem ser gerenciados ao longo do projeto
 - Esboço da arquitetura do sistema, que servirá como alicerce para a sua construção.

Fase 1 - Concepção

- Cada iteração está voltada para a produção de **casos de uso de negócio** e da **arquitetura preliminar**

Fase 2 - Elaboração

- O objetivo é **capturar** o que pode ser feito no sistema com as **restrições financeiras** e outras questões levantadas. Nessa fase se faz
 - **Captura dos requisitos funcionais** da aplicação (**que não foram capturados na fase de concepção**).
 - **Expansão do esboço da arquitetura** para uma arquitetura base para o desenvolvimento do sistema.
 - **Abordagem dos riscos significativos**.
 - **Elaboração de um plano** com **questões econômicas** indicando como o projeto vai evoluir na fase de construção.

Fase 2 - Elaboração

- Nesta fase, as iterações estão voltadas para a produção da **arquitetura básica**
- Vários dos casos de uso são especificados com **detalhes**
- A arquitetura do sistema é projetada

Fase 2 - Elaboração

- A arquitetura é expressa em visões dos modelos do sistema
 - modelo de casos de uso
 - modelo de análise
 - modelo de projeto
 - modelo de implementação
 - modelo de distribuição
- O resultado da fase é o conjunto dos modelos acrescidos de elementos de implementação

Fase 3 - Construção

- Nesta fase, as iterações estão voltadas para a produção de **módulos executáveis**, culminando com o desenvolvimento de um **produto pronto** para **entrega** à comunidade de usuários
- Durante esta fase o **produto é construído**
- O sistema torna-se um produto pronto para ser posto à disposição da comunidade de usuários

Fase 3 - Construção

- A arquitetura é estável, ainda que possa ser aperfeiçoada
- Ao final da fase, o sistema conterá todos os casos de uso que gerência e usuários concordaram em desenvolver para esta versão do produto
- Erros poderão (irão com certeza !) ser encontrados e serão corrigidos

Fase 4 - Transcrição

- Compreende o período em que o produto passa para **beta-teste**
 - a primeira versão é entregue aos usuários
- A fase envolve atividades de **treinamento de usuários, assistência de uso do produto** e **correção de defeitos** encontrados após a entrega
- Um pequeno número de usuários experientes utiliza o produto e reporta os erros e inadequações encontradas
- Os **desenvolvedores corrigem** os **erros** e incorporam melhorias

Requisitos

- Requisitos do sistema são especificados através da identificação das necessidades de usuários e clientes
- Estes requisitos funcionais são expressos em **casos de uso** através do **modelo de casos de uso**
- O modelo de casos de uso é desenvolvido em vários incrementos, onde as iterações irão adicionar novos casos de uso e/ou adicionar detalhes as descrições dos casos de uso existentes.

Fluxos do trabalho do processo - Requisitos

- Durante a **fase de concepção**, os **casos de uso mais importantes são identificados**, delimitando o domínio do sistema.
- Durante a **fase de elaboração**, a maioria dos **requisitos remanescentes é capturada**, assim desenvolvedores poderão saber o quão deverão se empenhar para desenvolver o sistema
 - Ao final da fase de elaboração, devem ter sido capturados e descritos cerca de 80% dos requisitos do sistema

Fluxos do trabalho do processo - Requisitos

- Porém, **apenas 5% a 10%** destes requisitos **devem ter sido implementados** na **fase de elaboração**
- Os **requisitos restantes** são **capturados** e **implementados** durante a **fase de construção**.
- Na **fase de transição** quase não há requisitos a serem capturados, a menos que ocorram mudanças nos mesmos.

Fluxos do trabalho do processo - Análise

- O modelo de análise fornece **mais poder expressivo e formalismo** como **diagramas de interações e diagrama de gráficos de estados** que representam a dinâmica do sistema.
- O fluxo de análise tem maior importância durante fase de elaboração
- Isso contribui para a **definição** de uma **arquitetura estável** e facilita o entendimento detalhado dos requisitos.

Fluxos do trabalho do processo - Projeto

- Objetivo é **suprir** as **necessidades especificadas pelos requisitos**
- **Descreve o sistema em um nível físico**
- Aqui obtem-se uma **compreensão detalhada** dos requisitos do sistema, levando em consideração fatores como linguagens de programação, SO, tecnologias de banco de dados, interface com o usuário, etc.
- Enfoque: fim da fase de elaboração -> início da fase de construção
- Serve de **base para o fluxo de implementação**

Fluxos do trabalho do processo - Projeto

- **Casos de uso são realizados por artefatos** de projeto representados por ferramentas de modelagem **também utilizadas no fluxo de análise** como diagramas de classes, diagramas de interação e diagramas de estados, agora com o intuito de capturar os **requisitos de implementação**
- Os mesmos diagramas citados são construídos em um **nível mais físico que conceitual** (exemplo: tipagem de variáveis)
- Enquanto que o **fluxo de análise** se interessa por “**o que o sistema deve fazer**”, o **fluxo de projeto** diz respeito a “**como os requisitos serão implementados**”

Fluxos do trabalho do processo - Implementação

- O fluxo de implementação é baseado no modelo de projeto
- Implementa o sistema em termos de componentes (código fonte, executável, etc).
- A maior parte da arquitetura do sistema é definida durante o projeto.

Fluxos do trabalho do processo - Implementação

- O modelo de implementação se limita
 - Planejar as integrações do sistema em cada iteração. O resultado é um sistema que é implementado como uma sucessão de etapas pequenas e gerenciáveis
 - Implementar os subsistemas encontrados durante o projeto
 - Testar as implementações e integrá-los, compilando-as em um ou mais arquivos executáveis, antes de enviá-los ao fluxo de teste.

Fluxos do trabalho do processo - Implementação

- Este fluxo tem maior importância na fase de construção e apesar de ter suas características próprias, a maior parte de suas atividades é realizada de forma quase mecânica, pelo fato das decisões mais difíceis terem sido tomadas durante o fluxo de projeto.
- O código gerado durante a implementação, deve ser uma simples tradução das decisões de projeto em uma linguagem específica.

Fluxos do trabalho do processo - Teste

- O fluxo de teste é desenvolvido com base no produto do fluxo de implementação.
- Os componentes executáveis são testados exaustivamente no fluxo de teste para então ser disponibilizados aos usuários finais.
- O principal propósito do fluxo de teste é realizar vários testes e sistematicamente analisar os resultados de cada teste.

Fluxos do trabalho do processo - Teste

- Componentes que possuírem defeitos retornarão a fluxos anteriores como os fluxos de projeto e implementação, onde os problemas encontrados poderão ser corrigidos.
- Um planejamento inicial de testes pode ser feito já na fase concepção.

Fluxos do trabalho do processo - Teste

- A maior parte dos testes são feitos durante a fase de elaboração, quando a arquitetura do sistema é definida, e durante a fase de construção quando o sistema é implementado.
- Na fase de transição, o fluxo de testes se limita ao conserto de defeitos encontrados durante a utilização inicial do sistema.

Fluxos do trabalho do processo - Teste

- O produto do fluxo de teste é o modelo de teste
- O modelo de teste pode descrever:
 - como componentes executáveis, provenientes do fluxo de implementação, são testados.
 - como aspectos específicos do sistema testados, como por exemplo, se a interface do usuário é útil e consistente ou se o manual do usuário cumpre seu objetivo.

Atividades de Fixação

Opa! Supresa !!!

Dúvidas?



Próxima Aula



Um pouco de história



Linux



Windows



MAC



Android / IOS



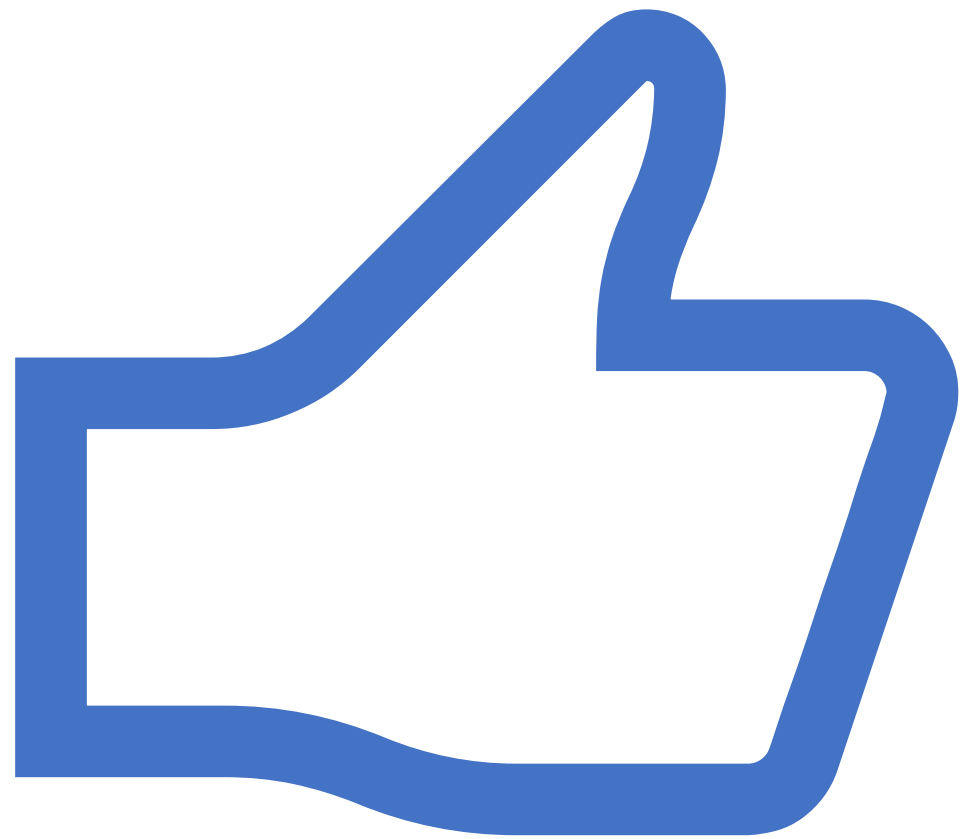
Futuras tendências



Dúvidas?



Obrigado !





Apresentador

Thales Levi Azevedo Valente

E-mail:

thales.l.a.valente@gmail.com