

# Processo de Desenvolvimento de Software

Prof. Luiz Leão

[luizleao@gmail.com](mailto:luizleao@gmail.com)

[luizleao.com](http://luizleao.com)

---

# Introdução

# Unidade I – Conceitos Gerais de Processo de Desenvolvimento de Software (PDS).

- O que é? Para que serve?
- Problemas mais comuns.

## Unidade II – Atividades em PDS

- Análise econômica e de requisitos.
- Especificação do Software.
- Desenho ou Arquitetura do Sistema de Software
- Codificação (Implementação)
- Teste do Produto

## Unidade III – Suporte e Manutenção do Software

- Documentação.
- Suporte e Treinamento
- Melhoria Continua.

## Unidade IV – Introdução aos padrões de PDS

- CMM / CMMI.
- ISO/IEC 12207.
- SPICE (ISO/IEC 15504).
- MPS/BR.

## Unidade V – Modelagem de PDS

- Processo Cascata (Water Fall) ou TOP DOWN.
- Processo Iterativo.
- Processo Ágil.

## Unidade VI – Processo Unificado

- Fases do Processo.
- Ciclo de vida do processo.



## Unidade VII – Ferramentas de PDS

- RUP (Rational Unified Process)
- PRAXIS



Vamos começar...

# Unidade I – Conceitos Gerais de Processo de Desenvolvimento de Software (PDS).

- O que é? Para que serve?
- Problemas mais comuns.

## O que é um Sistema?

- É um conjunto de procedimentos que estabelecem o funcionamento do fluxo de trabalho (Workflow) das organizações.
- Ex: Secretária que organiza o agendamento dos pacientes de um consultório médico.

## O que é um Software?

- É um conjunto de instruções que, orquestradas entre si, manipulam um conjunto de dados afim de produzir **informação** e **eventos**.
- Ex1: Sistema Web de agendamento de consultas médicas (Informação)
- Ex2: Sistema de Monitoramento de alerta de tempestades (Evento)

## Sistema vs *Software*

- O *software* é a automatização de um sistema.
- Nem todo sistema é automatizado.
- Ex: Agendamento manual de consultas médicas.

## Ainda sobre *software*...

- O *software* é tanto o produto quanto o veículo para entregar o produto (PRESSMAN, 2002)
- Ex: Um relatório gerencial em PDF é o produto, assim como o sistema usado na sua geração.

## Engenharia de *Software*

- “É a criação e a utilização de sólidos princípios de engenharia a fim de obter *software* de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais.”  
(Fritz Bauer)



## Evolução do *Software*

- No início, os *softwares* eram muito pequenos, dadas as limitações de *hardware*.
- Com o crescimento do poder computacional, cresce também o tamanho e a complexidade do *software*.

## Evolução do *Software* (Cont.)

- Várias técnicas surgiram para ajudar na administração dessa complexidade:
  - Técnicas ligadas à linguagens de programação;
  - Aprofundamento dos estudos na Engenharia de *Software*;
  - Arquitetura de *Software*;
  - Ferramentas CASE.

Tava tudo lindo até que...

## Crise do *Software*

- “A maioria dos especialistas concorda que o modo mais provável de o mundo ser destruído é por acidente. É onde nós entramos. Somos profissionais de computação. Nós causamos acidentes” (Nathaniel Borenstein)

## Crise do *Software*

- Anos 60, mas se arrasta até hoje
- Segundo pesquisa do Standish Group (2009)
  - 18% dos projetos são cancelados por atrasos e orçamentos estourados;
  - 52% dos projetos estouram o orçamento e/ou prazo;
  - 30% de todos os projetos de TI atingem seus objetivos dentro de prazo e custo estimados.

## Crise do *Software*

- Outros motivos:
  - Taxa de rotatividade de pessoal elevada – Entre 20% e 30% ao ano
  - Grandes sistemas levando de 3 a 5 anos para serem desenvolvidos . Muitos deles se tornando obsoletos antes de serem entregues – Natimortos

## Crise do *Software*

- Manutenção de *software* responsável pelo maior custo relacionado a computação para a maioria das empresas da área

## Crise do *Software*

- Segundo Sommerville (2007), os problemas a seguir foram definidos como desafios-chave a serem superados:
  - **Heterogeneidade** - Técnicas de desenvolvimento para construção de software que possam lidar com ambientes heterogêneos.
  - **Entrega** - Técnicas de desenvolvimento para conduzir a entrega mais rápida de *software*.
  - **Confiança** - Técnicas de desenvolvimento que mostram que o software pode ter a confiança dos seus usuários.

## Crise, mas nem tanto...

- Apesar dos problemas relatados persistirem de alguma forma, até hoje, vivenciamos muito mais casos de sucessos que insucessos.
- Alguns autores defendem que nunca existiu crise de *software*. Apenas uma Aflição Crônica (TEICHROW, 1989)
- Aflição: “Qualquer coisa que causa dor ou desconforto”.
- Crônica: “de longa duração ou que volta frequentemente”

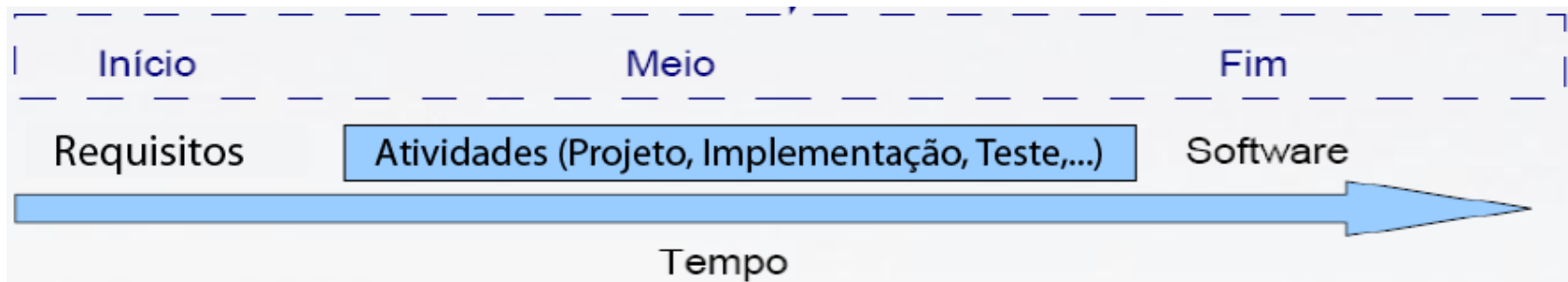


## Processo de Desenvolvimento de *Software* (PDS)

- É a estrutura comum, composta por um pequeno número de atividades, que são utilizadas em **todos** os projetos de *software* (PRESSMAN, 2002).
- Há muitos modelos para esses processos, cada um descrevendo abordagens diferentes para uma variedade de tarefas a serem executadas durante o processo.

## Ciclo de Vida e atividade de um processo

- Todo ciclo de desenvolvimento tem início, meio e fim.
- A variável temporal desse processo é chamada de **Ciclo de Vida**



## Ciclo de Vida e atividade de um processo (cont.)

- Todo processo é composto de atividades executadas, coordenadamente, num encadeamento de fases.
- Compreende todas das atividades relacionadas à definição, desenvolvimento, teste e manutenção do *software*.
- Existem vários processos. Cada um com a sua aplicabilidade.



Próxima Unidade...

Unidade II – Atividades em PDS