

Processo de Desenvolvimento de Software

Prof. Luiz Leão

luizleao@gmail.com

luizleao.com

Unidade II – Atividades em PDS

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

Atividade do PDS

- **1. Análise**
 - Investigar requisitos do usuário
- **2. Especificação**
 - Definir, claramente, as necessidades do sistema
- **3. Projeto**
 - Criar (ou adaptar) uma solução adequada
 - Definição de tecnologias e arquitetura a serem adotadas

Atividade do PDS

○ 4. Implementação

- Desenvolver a solução Proposta

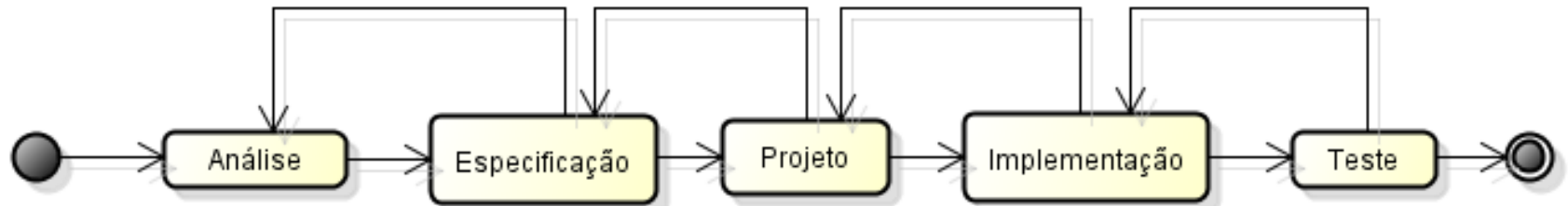
○ 5. Teste

- Garantir que a solução resolverá o problema em questão

○ 6. Manutenção

- Modificar a solução sempre que novos requisitos forem identificados

Atividade do PDS



Atividade do PDS

- Listagem de atividades mais comuns
- A adoção de cada uma delas depende das necessidades encontradas
- **Atividades complementares:**
 - Implantação
 - Homologação
 - Documentação
 - Treinamento
 - Etc.

1. Análise

- Mito: “O usuário não sabe o que quer”
- Verdade: “O cliente sabe o que quer, mas nem sempre o analista tem os instrumentos ou a expertise necessária para coletar os requisitos necessários para apontar uma solução”
- O analista deve se tornar um especialista no **negócio do cliente**, conhecendo assim o **domínio** do problema

1. Análise

- Artigo: Engenharia de Software: Uma história de amor e ódio – Kiko Reis
- <http://www.async.com.br/~kiko/FeiraUFSCar2003/0.php>
- Sobre o Engenheiro de Software:
 - “A profissão moderna mais difícil, de longe.”
 - “Requer bom planejamento e (paradoxalmente) experiência prévia.”

1. Análise

- **Fases:**

- **1. Descoberta:**

- Escutar e observar

- **2. Refinamento:**

- Interrogar e esclarecer

- **3. Modelagem:**

- Sugerir e verificar

1. Análise

- Com base nas informações coletadas, analisamos, previamente, a relação:

Bolso do Cliente x Custo do Produto

- Métricas de software (Kloc, Ponto Função, etc.) são bem vindas para nortear a viabilidade do projeto

1. Análise

- **Fatores geradores de custo do software:**
 - Processo (Treinamento, gestores, ...)
 - Pessoal
 - Tempo
 - Ferramentas
 - Ambiente (Servidores, SGBD, ...)

1. Análise

○ **Tempo: Inimigo x Amigo?**

- Deve ser dimensionada a necessidade do cliente, com a capacidade de produção da equipe
- Menos Tempo → Mais Custo
- Nem sempre o cliente está disposto a investir

1. Análise

- **Artefatos de entrega (Saída):**
 - Informações necessárias para a elaboração do documento de especificação de requisitos, de preferência, em uma **Minuta** padrão para essa finalidade
- **Ferramentas**
 - Questionário de entrevista
 - Etc.
- **Possíveis Falhas**
 - Induzir o cliente à utilizar uma solução, sem ter todas as perguntas respondidas, pode gerar um escopo de produto que não o atende

2. Especificação

- Última fase da tarefa de análise
- É necessário que seja escrito de forma **não ambígua** qual é o **comportamento requerido**
 - Notações formais
 - Documentos estruturados
 - Exemplos
- **Artefatos de Entrega (Saída)**
 - Especificação de requisitos que comunique ao projetista as características requeridas para o sistema, usando **Casos de Uso Descritivos**
 - Diagramas de Casos de Uso

2. Especificação

○ **Possíveis Falhas**

- Documentação insuficiente para o entendimento do problema
 - Retrabalho
 - Ocasiona atraso na execução das fases posteriores

3. Projeto

- Desenhar uma solução que seja **fiel aos requisitos**
 - Com base na experiência acumulada (e técnicas padronizadas)
- Geralmente precisam **inovar** em um certo nível
- É possível que surjam várias soluções
 - Recomenda-se o uso de alguma métrica para a escolha da melhor solução para o projeto

3. Projeto

- Fase de definição de diversas tecnologias:
 - Linguagem de Programação
 - Padrões de Projetos
 - Materializados na adoção de frameworks
 - IDE
 - Banco de Dados
 - Etc.

3. Projeto

○ Artefatos de Entrega (Saída)

- Um documento de projeto que, de forma **não ambígua** e **clara**, comunica o projeto com a equipe que irá implementar o software.

Recomenda-se que contenha:

- Concepção arquitetural
- Ferramentas a serem adotadas
- Frameworks
- Etc.

3. Projeto

- **Artefatos de Entrega (Saída) (cont.)**
 - Diagramas Estruturais
 - Diagrama de Classes
 - Diagrama de Componentes
 - Etc.
 - Diagramas Comportamentais
 - Diagrama de Atividades
 - Diagrama de Sequência
 - Diagrama de Máquina de Estados
 - Etc.
 - Modelagem de dados

3. Projeto

- **Possíveis Falhas:**

- Arquitetura do software difícil de ser replicada para a equipe
 - Curva de aprendizagem desfavorável

4. Implementação

- **Atividades relacionadas ao código-fonte do sistema:**
 - Escrever
 - Documentar
 - Debugar
 - Preparar a aplicação para testes
 - Unitário
 - Usuário

4. Implementação

- **Artefatos de Entrega (Saída)**
 - Código-fonte, de preferência, versionado
 - Documentação do código-fonte
 - Produto para teste

4. Implementação

○ **Possíveis Falhas:**

- Implementação em desacordo com o requisito.
- Descumprimento de prazo
 - Dimensionamento equivocado do esforço a ser colocado nas tarefas de implementação
 - Déficit técnico

5. Teste

- Fase de verificação das funcionalidades implementadas
- **Missão:**
 - Checar se a implementação **corresponde ao projeto** (e se funciona)
 - Checar se a implementação **atende aos requisitos**
- Efetuadas as verificações, o Analista de Teste vai elaborar um **Plano de Teste**

5. Teste

- **Plano de Teste:**

- Conjunto de procedimentos, cuja finalidade é de validar os artefatos de software produzidos até então
- A definição da unidade a ser testada no plano fica sobre responsabilidade do **Analista de Teste:**
 - Módulo
 - Tela
 - Requisito
 - Etc.

5. Teste

- **Artefato de Entrega (Saída):**
 - Plano de Teste
 - Resultado da execução do plano de teste, sinalizando as unidades de teste que foram bem sucedidas e as que apresentaram problemas

5. Teste

- **Artefato de Entrega (Saída):**
 - Plano de Teste
 - Resultado da execução do plano de teste, sinalizando as unidades de teste que foram bem sucedidas e as que apresentaram problemas

5. Teste

- **Possíveis Falhas:**

- Relação **Cobertura x Precisão:**

- Há uma tendência de se analisar as unidades de teste de forma superficial, para acelerar a entrega, aumentando a **cobertura** do teste, diminuindo a sua precisão.
- Implementações equivocadas do software acabam passando despercebidos aos olhos do Analista de Teste, principalmente se tiverem uma natureza sutil

Exercícios

1. Qual a diferença entre um software e um sistema? Qual dos dois conceitos é mais amplo? Por quê?
2. O que você entende por processo de desenvolvimento de software e qual a sua importância para a qualidade dos produtos de software?
3. Qual a diferença entre processo e projeto de software?
4. O que é domínio da aplicação (ou do problema)?
5. Suponha que você iniciou suas atividades em uma empresa e lhe é solicitado desenvolver um software. Quais as suas primeiras providências, antes de colocar em prática suas habilidades de desenvolvedor?