

Software: "Conjunto de programas, procedimentos, regras e documentação associados a um sistema computadorizado"

#### 04 Características de um Software: Manutenibilidade, Confiabilidade, Eficiência, Usabilidade. 04 Elementos de Engenharia de Software: Processo, Ferramenta, Método, Qualidade

Mitos: 1 Ferramentas potentes x máquinas velozes = Software moderno e pronto  
2 Prazo curto x Software inacabado = Contratar mais programadores  
3 Todos programadores são iguais  
4 O programa/sistema está 95% pronto!

Verdade: 1 A equipe está treinada para usar as ferramentas? A equipe é competente?  
2 Prazo curto: é melhor negociar o prazo do que contratar novos programadores, devido ao problema de ambientação que os novos terão para se adaptar.  
3 Programadores tem habilidades diferentes  
4 Programadores são otimistas/confiança excessiva

#### Crise do Software

Problemas na manutenção do software  
Prazos e custos mal calculados  
Baixa qualidade dos produtos  
Pouca (ou nenhuma) documentação  
Baixa adequação às necessidades do cliente

#### Definição de Engenharia de Software

"Uso de métodos, ferramentas e técnicas para criar, gerenciar e evoluir produtos de software"

Não é possível programar sem se "programar"

### DISCIPLINAS BÁSICAS

#### REQUISITOS

- Levantar necessidades e enxergar oportunidades de melhorias
- Software tem que agregar valor, pois nem sempre vale a pena desenvolver software
- Quanto se vai ganhar ou economizar com o software e quanto vai custar?
- Que riscos de negócio um software minimiza?

#### ANÁLISE & PROJETO

Divisão do software em "pedaços lógicos" (módulos, subsistemas e componentes)

- Critério de divisão: por processos e regras de negócio inter-relacionadas
- Projeto de interface com usuário
- Telas, padrões, componentes visuais

- Projeto de integração com outros sistemas
- Formas de integração, tecnologias, etc.

Mapeamento dos riscos tecnológicos e de negócios

- Classificação
- Como resolvê-los
- Avaliação e aplicação de tecnologias
- Possibilidades de reuso

## **IMPLEMENTAÇÃO**

- Programação do código fonte
- Testes feitos pelo programador
- Padrões de programação

## **TESTES**

O software faz mesmo o que ele se propõe a fazer?

- Testes feitos pelo programador não garantem
- Usuário -> Projetista -> Programador
- Possíveis defeitos por conta das diferentes interpretações
- Programador tem visão limitada aos módulos que ele programou

Verificação de atendimento funcional às necessidades do cliente

- Padrões visuais
- Carga / demanda
- Homologação com o usuário

## **IMPLANTAÇÃO**

- Ambiente
- Treinamento
- Suporte
- Entrega e acompanhamento

Montar ambiente de execução

- Instalar o software
- Treinar o usuário
- Realizar eventuais migrações iniciais de dados e de processos
- Montar equipe de suporte técnico e funcional

- E...finalmente entregar o software e disponibilizar o mesmo para uso

## **MANUTENÇÃO CORRETIVA**

Pode haver defeitos não detectados nas etapas anteriores

- Software então pode sofrer alterações e ter novas versões por conta de correções

## **MANUTENÇÃO EVOLUTIVA**

Um software evolui com as necessidades de negócio e tecnológicas.

- Software então pode sofrer alterações e ter novas versões por conta de evoluções e de alterações demandadas pela própria evolução e alteração das necessidades dos clientes

## **DISCIPLINAS DE SUPORTE**

### **GERÊNCIA DE PROJETOS**

Gestão de pessoas, cronograma, custos, qualidade

### **AMBIENTE**

Infra-estrutura, Ferramentas, Máquinas, Rede e Internet

### **GERÊNCIA DE CONFIGURAÇÃO E MUDANÇAS**

Requisitos, Versões do software (releases e patches), custos e prazos

Ferramentas Case são todas as ferramentas que dão suporte a Engenharia de Software (Para construção do código, testes unitários, controle de versão do código fonte, ex: Eclipse, IDE's e Visual Studio voltado para c#, visual basic, c++).

IDE's de desenvolvimento são softwares capaz de prover um ambiente para escrita do código, tem um sofisticado editor de texto, compilador e executor integrado e também faz checagem de padrões de codificação.

## **Exemplos sobre TESTES**

Teste de sanidade: Não quer encontrar defeito, visa verificar o funcionamento básico

Teste de sistema: Verifica se o Software apresenta defeitos críticos ou graves

Teste de regressão: verifica se após a correção, o Software apresenta defeitos novos

Teste de aceitação (Homologação): É a validação do Software pelo o usuário final

Teste funcional (Regras de negócio): É feito um teste baseado na regra de negócios

## **OS QUATRO MODELOS DE DESENVOLVIMENTO DA ENG. DE SOFTWARE:**

CASCATA

EVOLUCIONÁRIO

REUSO

FORMAL

## **ITERAÇÃO DE PROCESSOS**

Híbridos: Combinam dois modelos de processo

Incremental: Modelo intermediário Ex: Cascata <-> Evolucionário  
Espiral: Gerenciamento de risco Explícito  
METAPROCESSOS

RUP : Projetos grandes e complexos que exigem documentação elaborada ou por força da lei  
Em Rup não se fala em Engenharia de Software e sim de WorkFlows (Fluxo de Trabalho)

Fases do RUP: Concepção, Elaboração, Construção e Transição

XP + Scrum: Projetos pequenos e médios, sem muita documentação  
Processo ágil

Fases do XP: Exploração, Planejamento, Iterações para Release, Validação para Produção, Manutenção e Morte.

**Guidelines são regras, recomendações, heurísticas que suportam atividades e passos.**

PROCESSO ÁGIL: Tem a finalidade de agilizar os processos, dividindo interações, diminuindo os riscos e ir apresentando ao cliente.

SCRUM é um processo interativo auto-gerenciável que teoricamente pode ser aplicado a qualquer processo de trabalho. É uma forma para desenvolvimento equipes e indivíduos.

Em Scrum não se fala em reunião e sim cerimônias  
Em Scrum não se fala de requisitos e sim histórias

Product Owner é o indivíduo responsável que vai priorizar os processos  
Scrum Master ele não é um gerente, e sim responsável pela equipe e o bem estar  
Equipe é responsável pela entrega dos itens