Teste de Software

Nivelamento

PUC Minas

Instituto de Ciências Exatas
e Informática

Prof. Lesandro Ponciano

Departamento de Engenharia de Software
e Sistemas de Informação (DES)

Objetivos da Aula

- Analisar conceitos associados aos softwares
  - Definições, tipos, defeitos
- Discutir o desenvolvimento de software
  - Bem-sucedido e malsucedido
  - Êxito no desenvolvimento de software
  - Surgimento da engenharia de software
- Contextualizar testes na Engenharia de Software
  - Atividades

Software

```
private void btnOpen_Click(object sender, EventArgs e)
{
    string file_name = "C:\\test1.txt";
    string textLine = "";

    System.IO.StreamReader objReader;
    objReader = new System.IO.StreamReader(file_name);

    do
    {
        textLine = textLine + objReader.ReadLine() + "\r\n";

    } while (objReader.Peek() != -1);

    textBox1.Text = textLine;

    objReader.Close();
}
```

Softwares

… são elementos do sistema lógico, não do sistema físico

… são instruções (programas de computador) que quando executadas fornecem características, funções e desempenho desejados

… incluem estruturas de dados que possibilitam aos programas manipular informações adequadamente

… incluem informações descritivas sobre a operação e uso dos programas

Produtos de Software

- Produtos de software podem surgir da necessidade de um usuário ou de um mercado

- Produtos sob encomenda (personalizados)
  - Ex.: Sítios web, controles de processos específicos de uma organização

- Produtos genéricos (de prateleira, *stand-alone*)
  - Ex.: Editores de texto, pacotes gráficos, ferramentas de gerenciamento de projetos, jogos

Tipos de Software

- O tipo do software determina
  - Interfaces para usuários e proximidade com o cliente
  - Funcionalidades, eficiência e custo da implementação

- Alguns tipos de software
  - Software de sistema e de aplicação
  - Software de engenharia ou científico
  - Software embutido (ou embarcado)
  - Software de linhas de produto
  - Aplicações para Web
  - Software de inteligência artificial

Algoritmos

... são “descrição de um conjunto de comandos que, obedecidos, resultam numa sucessão finita de ações” (Farrer et al. 1999)

... são “Sequência de passos computacionais que transformam uma entrada em uma saída” (Cormen et al. 2002)

... são "um processo sistemático para a resolução de um problema". (Szwarcfiter et al. 2010)

Estrutura de Dados

- É a forma de armazenamento e organização de dados que é usada para resolver o problema
  - A estrutura pode afetar a eficiência da solução

- Existem estruturas de dados estáticas e dinâmicas
  - Estruturas de dados estáticas são compostas por um número fixo (finito) de elementos
  - Estruturas de dados dinâmicas podem ter um número indefinido de elementos

Softwares Grandes

- Máquinas de busca, como Google
  - Algoritmos como *page rank* e *map-reduce*
- Redes sociais, como Facebook
  - Algoritmos de processamento de grafo
- Sistemas Peer-to-Peer (P2P), como BitTorrent
  - Algoritmos baseados em reputação como o “olho por olho dentes por dentes” (*tit-for-tat*)

Desenvolvimento de Software

Desenvolvimento de Software

- Desenvolver um software significa
  - Transformar a necessidade de um usuário ou de um mercado em um produto de software
  - Elaborar e implementar um sistema computacional

- Um software pode ser bem-sucedido ou malsucedido

Software

Bem-sucedido

- Atende às necessidades dos usuários
- Opera perfeitamente durante um longo período
- É fácil de modificar
- É fácil de utilizar

Malsucedido

- Seus usuários estão insatisfeitos
- É propenso a uma diversidade de defeitos
- É difícil modificá-lo
- É difícil utilizá-lo

Engenharia de Software

- “é 1) aplicação de uma abordagem sistemática, disciplinada, e quantificável no desenvolvimento, na operação e na manutenção de software, 2) estudo e abordagem do descrito em 1” (IEEE*, 1993)

*IEEE é o Instituto de Engenheiros Eletricistas e Eletrônicos (*Institute of Electrical and Electronics Engineers)*

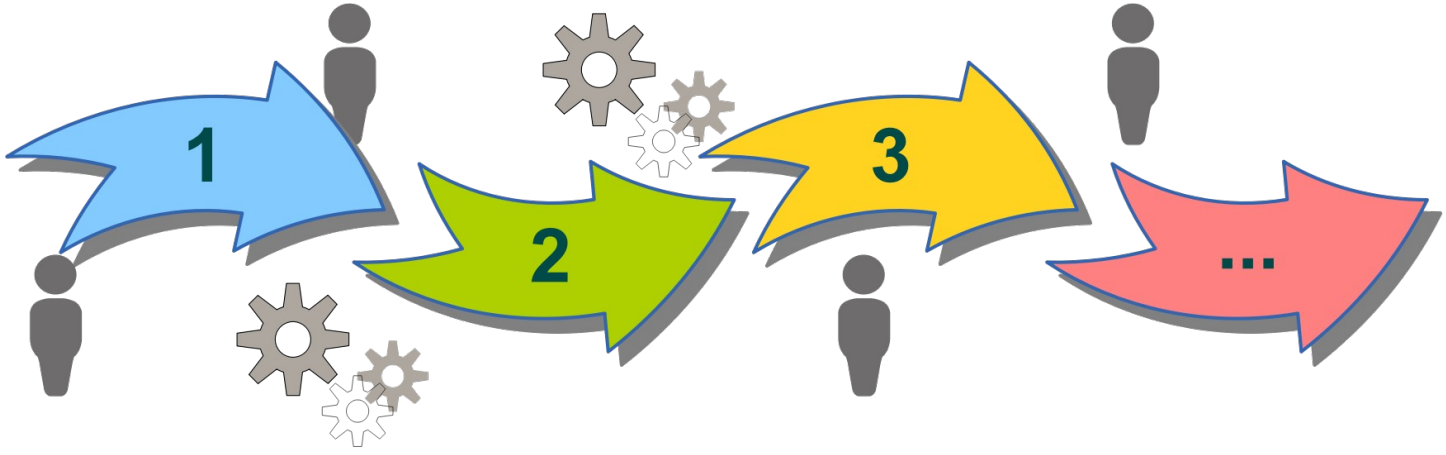
Processo de Desenvolvimento

- Desenvolver um software significa
  - Transformar a necessidade de um usuário ou de um mercado em um produto de software
  - Elaborar e implementar um sistema computacional

- Processo de Desenvolvimento de Software, ou simplesmente
  - Processo de Desenvolvimento
  - Processo de Software

Processo de Software

- Um processo de software consiste de um conjunto de atividades realizadas por pessoas e cujo objetivo é o desenvolvimento ou evolução de um artefato, que é o software e sua documentação



Papel, Atividade e Artefato

- Conceitos básicos
  - Papel é uma função desempenhada por um membro da equipe de desenvolvimento
  - Atividade é uma tarefa realizada por uma pessoa em um determinado papel a fim de gerar um determinado artefato
    - Atividades consomem e produzem artefatos
    - Estão sujeitas a pré-condições e pós-condições
  - Artefato é o produto de uma atividade, como um modelo, documento, código

Atividades Genéricas

- Pressman (2011)
  - Comunicação
  - Planejamento
  - Modelagem
  - Construção
  - Emprego
- Sommerville (2011)
  - Especificação
  - Projeto e implementação
  - Validação
  - Evolução

Especificação

Projeto e Implementação

Validação

Evolução

Atividade de Especificação

- Requisitos de Software
  - O que o software deve fazer, serviços que deve oferecer, restrições ao seu funcionamento

- Ou, ainda..
  - Condição ou capacidade necessária a um usuário para resolver um problema ou alcançar um objetivo
  - Condição ou capacidade que deve ser alcançada ou possuída por um sistema para satisfazer um contrato
  - Uma representação documentada de uma condição ou capacidade como nos itens 1 ou 2

Requisitos Funcionais

- Requisitos funcionais (RF) são declarações de
  - Serviços que o sistema deve oferecer
  - Como o sistema deve reagir a entradas específicas
  - Como o sistema deve se comportar em determinadas situações

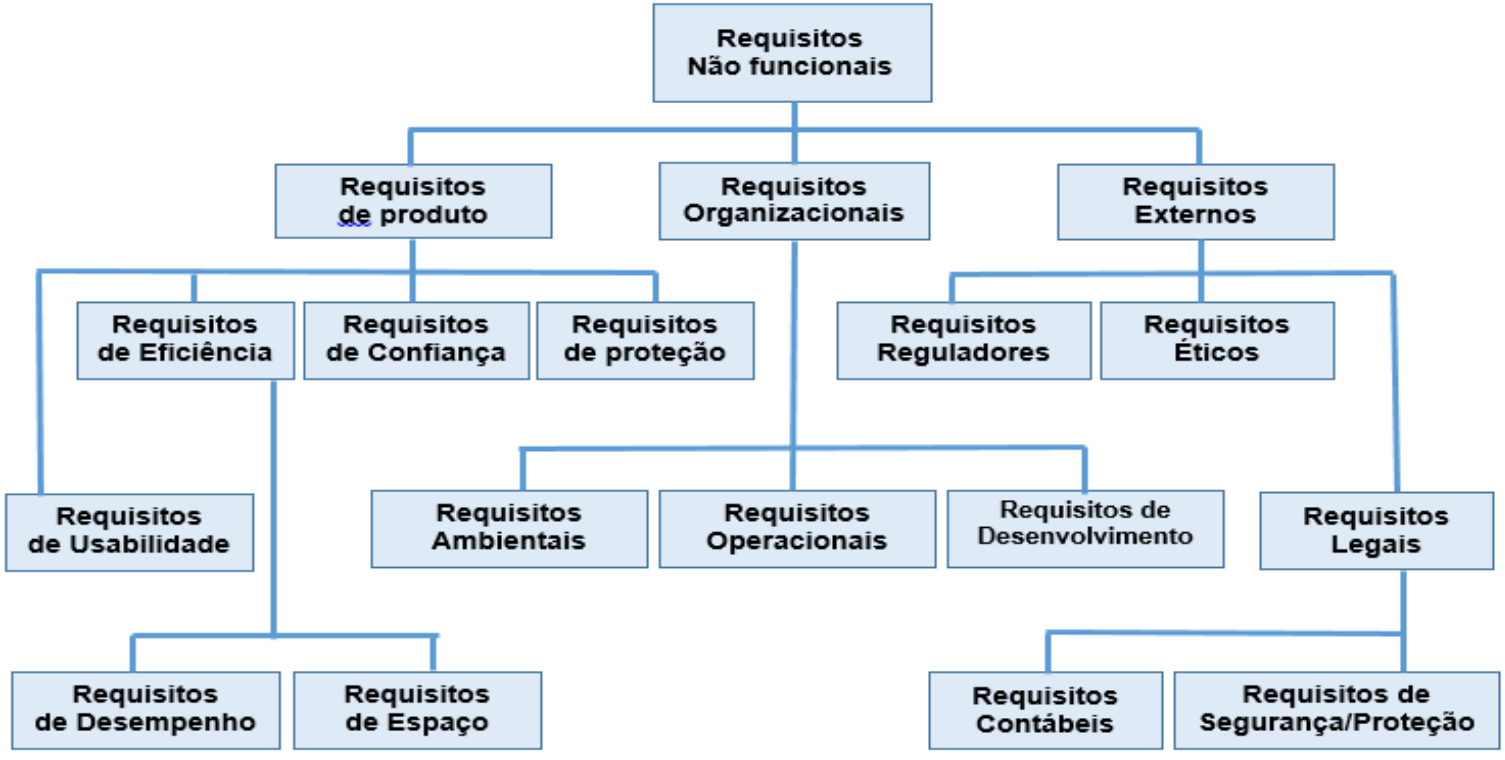
- Estão diretamente relacionados aos objetivos do sistema

Requisitos Não-Funcionais

- Requisitos não-funcionais (RNF),
  - Restrições aos serviços ou funções oferecidos pelo sistema, como tempo (*timing*), processo de desenvolvimento, plataforma, restrições legais

- Surgem de restrições de orçamento, políticas organizacionais, necessidade de interoperabilidade e fatores externos

Tipos de Requisitos Não-Funcionais



Atividade de Projeto e Implementação

- O software deve ser produzido para atender às especificações

- O projeto (*design*) é uma extensão da especificação visando a implementação em um computador
  - É algo a ser compreendido pelo programador
  - O artefato é uma "planta", "desenho" ou “modelo"

- A implementação é a escrita do código
  - Atividade simples se o projeto for bem-feito
  - O artefato é o código equivalente ao projeto

O Artefato da Atividade Projeto

- Desenhos, ou plantas que descrevem a forma de implementação de um conjunto de requisitos
  - Diagramas em alguma linguagem representacional
  - Quais diagramas?
    - Depende da linguagem (ex. UML)
    - Depende do que processo determina
    - Depende do software que se está projetando

Projeto-Implementação

- É difícil separar "Projeto" de "Implementação"
- Um projeto está relacionado à estrutura geral e específica do software
  - Projeto da arquitetura
  - Projeto de Interface
  - Projeto de componentes
  - Projeto de Banco de Dados
- Projeto pode ser representado de diversas formas, mas sua materialidade é o código fonte

Model-Driven Development (MDD)

- Engenharia Dirigida a Modelos
  - Modelos de software são criados em diferentes níveis de abstração
  - Modelos estruturados

- Modelos com detalhes suficientes para que um sistema executável possa ser gerado a partir dele

- Ferramentas de desenvolvimento de software podem ser usadas para gerar um “esqueleto” do código a partir do projeto

Atividade de Validação

- O software deve ser validado para se certificar de que ele atende os requisitos
  - Mostrar que o software se adéqua às especificações
  - Enquanto atividade genérica, inclui “verificação e validação” / “teste de software”

- Inclui várias fases de teste do software
  - Testes feitos pelo próprio programador durante a programação
  - Testes feitos por equipes independentes
  - Testes manuais e automatizados

Evolução

- O software deve evoluir para atender às necessidades de mudanças dos clientes

- A evolução inclui
  - Correção de defeitos
  - Inclusão de novas funcionalidades
  - Adaptação para novas plataformas

Processos Dirigidos a Planos

- As atividades são “planejadas com antecedência"
  - Processos e modelos tradicionais/convencionais
- Trabalha com ideia de previsibilidade
  - A evolução é medida em relação ao plano
- Dinâmica
  1) especificar completamente os requisitos
  2) projetar completamente o sistema
  3) construir todo o sistema
  4) testar todo o sistema implementado

Processos baseados em Métodos Ágeis

- Origem a partir da metade de 1990
- Acomodar mudanças e tratar o *overhead* no planejamento, projeto e documentação do sistema
  - Trabalha com a ideia de adaptabilidade
- *Overhead* de documentação é injustificado em
  - Sistemas para empresas de pequeno porte
  - Sistemas que serão alterados rapidamente em função da dinâmica do negócio
  - Sistemas que serão mantidos por pouco tempo
    - Manutenção inexistente ou pouca

Exercício de Fixação

- O que é um software?
- Quando o desenvolvimento de software é considerado bem sucedido?
- Qual a principal diferença entre a origem de defeitos em software e a origem de defeitos em hardware?
- Quais são as atividades genéricas de engenharia de software?

Referências

- SOMMERVILLE, Ian. Engenharia de Software - 9a edição. Pearson ISBN 9788579361081. (Capítulo I e 2)

- PRESSMAN, Roger. Engenharia de software. 8. Porto Alegre ISBN 9788580555349. (Capítulo I e 2)