

Processos e Qualidade de Software

Nivelamento de Engenharia de Requisitos e Fundamentos de SI



PUC Minas

Instituto de Ciências Exatas
e Informática

Prof. Lesandro Ponciano

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

Objetivos da Aula

- Revisar conceitos de Engenharia de Requisitos e de Fundamentos de Sistemas de Informação
- Contextualizar os Processos de Desenvolvimento de Software
- Contextualizar a Qualidade de Software

O Caminho Até Aqui

- Processos e Qualidade de Software
 - Pré-Requisito: Engenharia de Requisitos
 - Pré-requisito: Fundamentos de Sistemas de Informação

Sistemas de Informação

- Sistema
 - Conjunto de elementos inter-relacionados que estão articulados e formam um todo organizado

- Sistema de Informação
 - Conjunto de componentes inter-relacionados trabalhando juntos para coletar, recuperar, processar, armazenar e distribuir informações
 - Componentes principais
 - hardware, software, redes e instalações de comunicação, banco de dados e pessoas

Hardware

- Armazenamento
- Processamento
- Transmissão de dados
- Autonomia
- Mobilidade



Pessoas

- Interessados (*stakeholders*), exemplos:
 - Acionistas
 - Clientes
 - Usuários
 - Mantenedores
- Fornecedores do software
 - Entendem o que o cliente quer e o que os usuários precisam
 - Especificam, projetam, implementam e validam o software
 - Mantêm e evolui o software ao longo do tempo

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
- Tipos de software
 - Software de sistema
 - Software 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)

A solução de um problema computacional

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*)

Diversos componentes, alta complexidade

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)

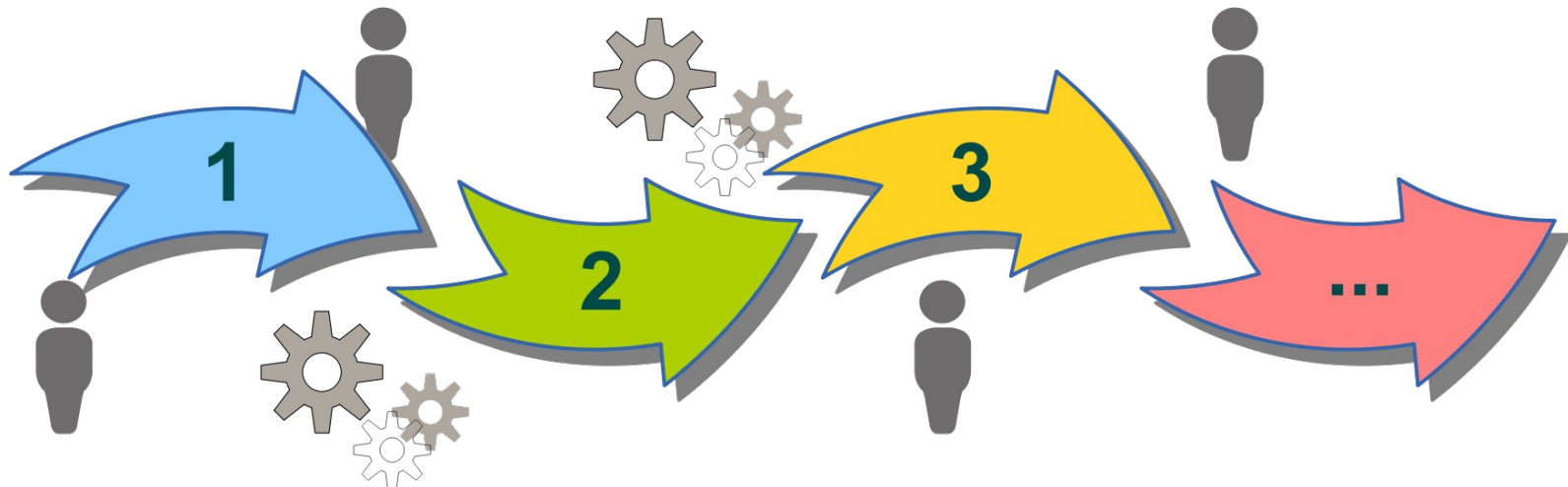
*IEEE é o Instituto de Engenheiros Eletricistas e Eletrônicos (*Institute of Electrical and Electronics Engineers*)

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



Processo de Software

- 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

Especificação: Requisitos do Software

- Requisitos são as descrições de
 - O que o software deve fazer
 - Os serviços que deve oferecer
 - As 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

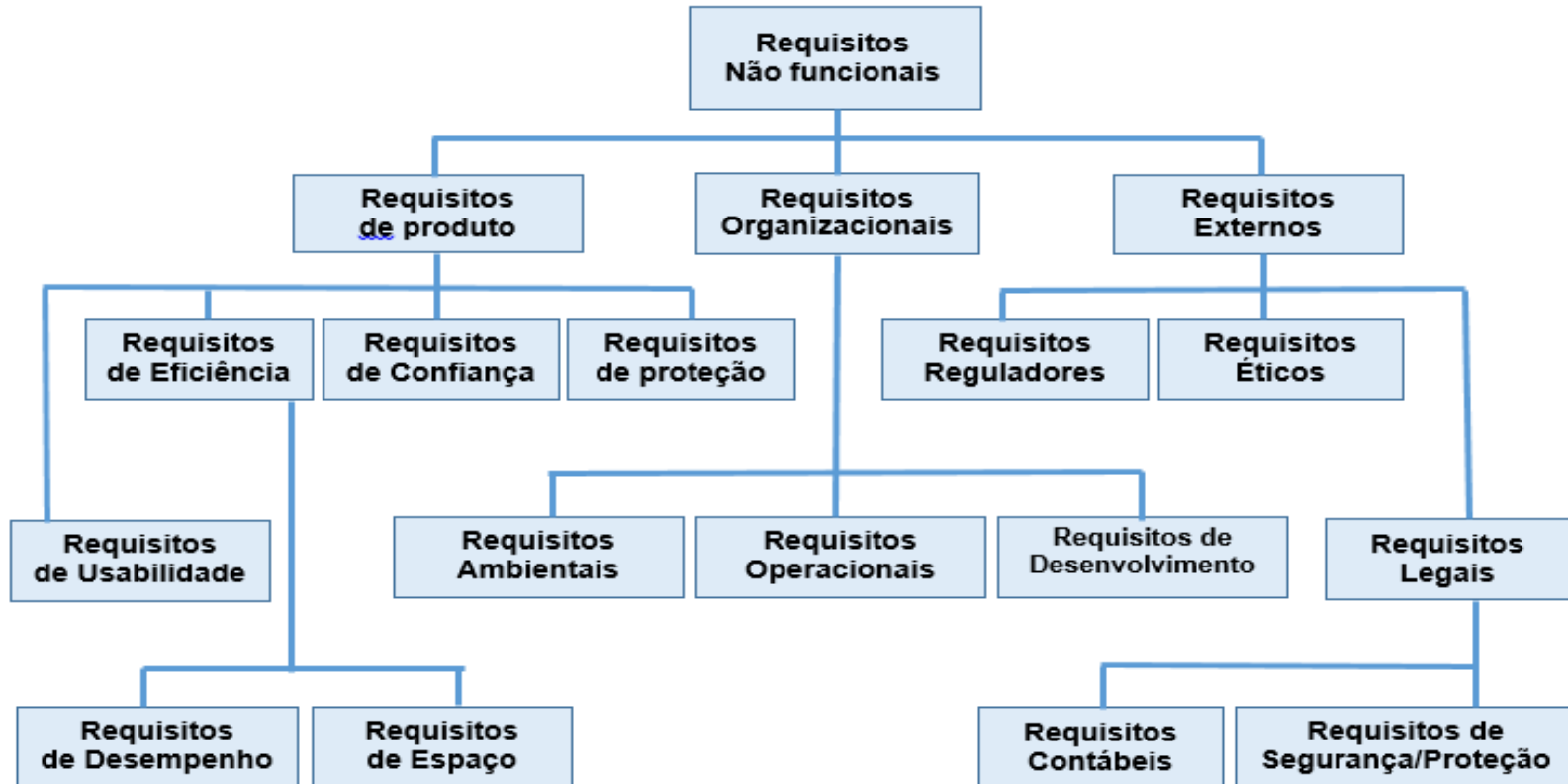
Requisitos Funcionais

- 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

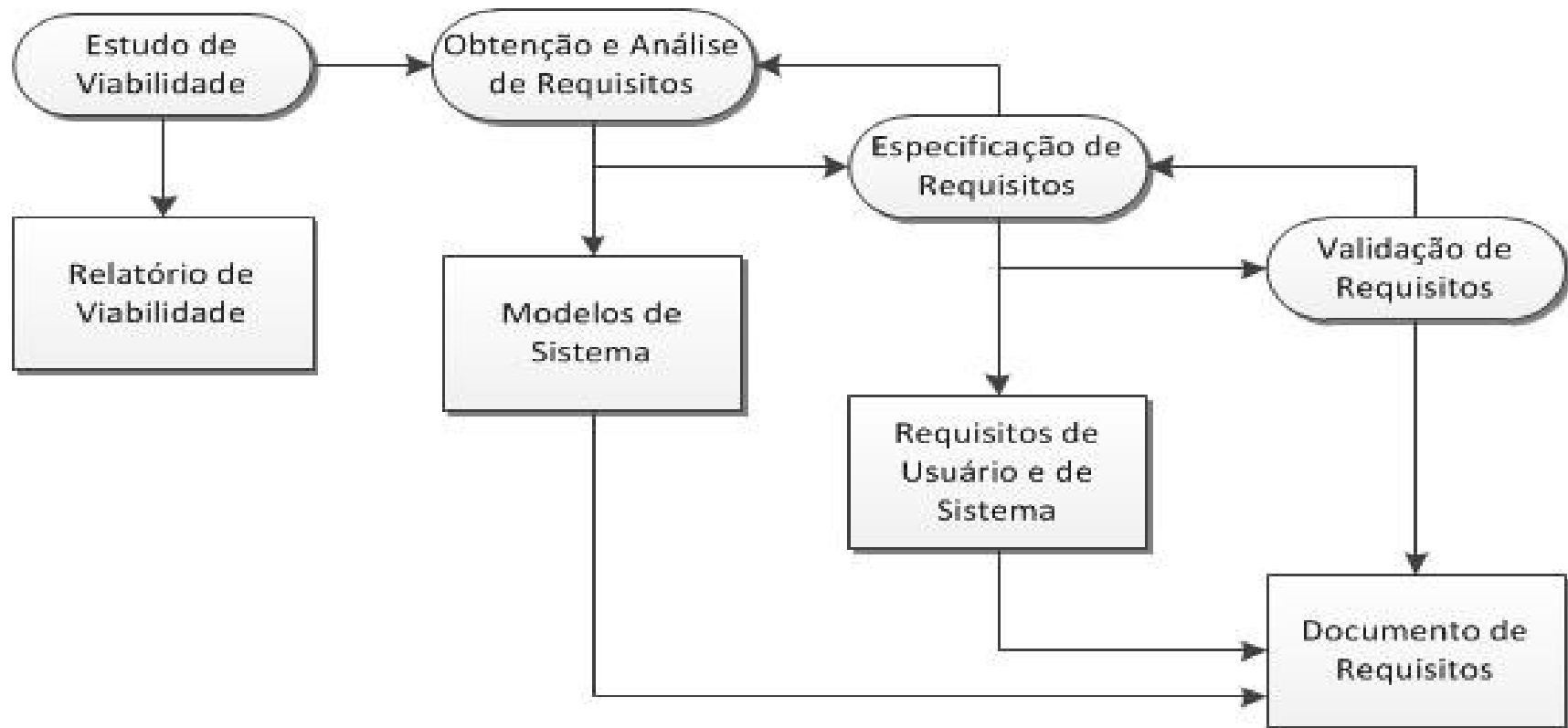
Requisitos Não-Funcionais

- Requisitos não-funcionais (RNF), são **restrições** aos serviços ou funções oferecidos pelo sistema, como
 - Restrições de tempo (*timing*)
 - Restrições de processo de desenvolvimento
 - Restrições impostas por plataforma
 - Restrições legais
- Surgem de restrições de orçamento, políticas organizacionais, necessidade de interoperabilidade e fatores externos

Tipos de Requisitos Não-Funcionais



Engenharia de Requisitos



Documento de Viabilidade

- Usado para ajudar na decisão se deve ir adiante ou não com um projeto (*project*) de sistema de informação
 - Deve-se ou não continuar?
 - Qual alternativa de sistema é mais viável?
- Análise de viabilidade
 - Viabilidade operacional
 - Viabilidade técnica
 - Viabilidade de cronograma
 - Viabilidade econômica

Documento de Visão

- Definição do escopo do sistema de alto nível
- Definição do propósito do programa, produto ou projeto
- Introduzir
 - o problema
 - a solução proposta
 - os recursos de alto nível do produto]
- Estabelecer expectativas e reduzir riscos

Documento de Requisitos

- Documento formal
 - Segue uma série de regras de escrita
 - Reflete um acordo ou contrato entre clientes, analistas e desenvolvedores
 - O que está definido nele é o que deve ser feito
- Elemento de comunicação
 - Todos conhecem os termos, regras, propriedades do sistema
 - Equipes usam ele para transmitir informações
- É produzido na Engenharia de Requisitos

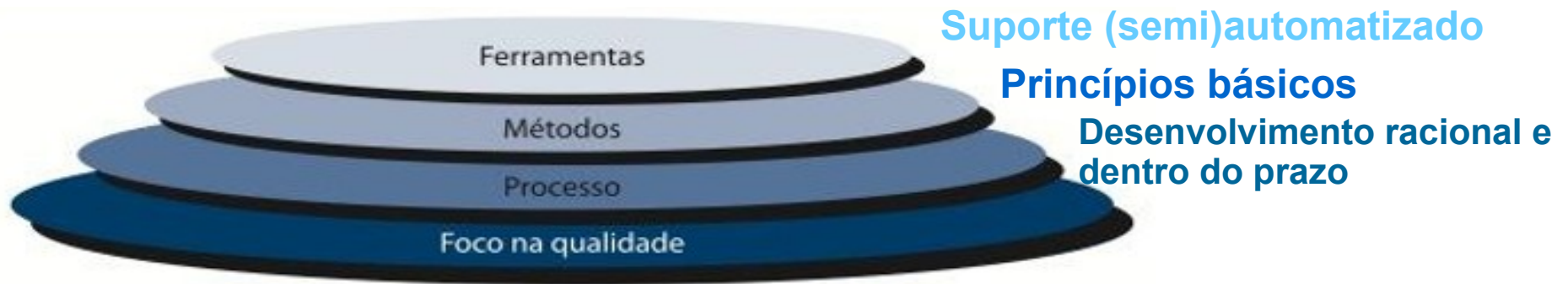
Validação de Requisitos

- Mostrar que os requisitos realmente definem o sistema que o cliente deseja
- Descobrir problemas na forma como os requisitos estão especificados
- Questões de Verificação
 - Validade: Quais serviços realmente são necessários?
 - Consistência: Existe conflito entre requisitos?
 - Completude: Todos os requisitos estão documentados?
 - Realismo: Os requisitos podem ser implementados?
 - Facilidade de verificação: Como verificar se o requisito foi implementado

Implementação

- O software deve ser produzido para atender às especificações
- Na Engenharia de Software, o projeto é uma extensão da especificação
- O projeto visa a implementação em um computador
 - É algo a ser compreendido pelo programador
- A implementação é a escrita do código
 - É uma tarefa relativamente simples se o projeto for bem-feito

Qualidade na Engenharia de Software



- A pedra fundamental que sustenta a engenharia de software é o foco na qualidade
- Qualquer abordagem de ES deve estar fundada em um comprometimento com a qualidade

Qualidade

- Pode-se falar em **qualidade do produto de software** e **qualidade do processo** de desenvolvimento do software
- Como saber se algo tem qualidade ou não?
 - Indicadores, características ou atributos de qualidade

Garantia e Controle de Qualidade

- **Garantia** de qualidade
 - Estabelecer procedimentos e padrões organizacionais que conduzam a um software de qualidade
- **Controle** de qualidade
 - Uma série de inspeções, revisões e testes usados ao longo do processo de software para garantir que cada produto de trabalho tenha qualidade

O que está por vir?

- Processos Dirigidos por Planos e Processos Ágeis
 - XP e Scrum
- Conceitos de modelos de maturidade
 - ISO, CMMI, MPS.BR
- Métricas e GQM
 - Estimativas de Tamanho, Esforço, Recursos, Cronograma
 - Pontos de Função
- Qualidade
 - Padrões IEEE para Qualidade de Software
 - Técnicas para Garantia da Qualidade
 - Procedimentos para Gestão da Qualidade

Exercício de Fixação

- Como algoritmos e estrutura de dados se relacionam com o projeto de sistemas de informação?
- Qual a importância da Engenharia de Requisitos para o projeto de sistemas de informação?
- O que é UML e para que ela serve? Qual sua importância na atividade de projeto de sistemas de informação?

Referências

- SOMMERVILLE, Ian. Engenharia de Software - 9a edição. Pearson ISBN 9788579361081. (Capítulo 1 e 2)
- PRESSMAN, Roger. Engenharia de software. 8. Porto Alegre ISBN 9788580555349. (Capítulo 1 e 2)