

# **REQUISITOS DE SISTEMAS**

## **APRESENTAÇÃO**

## **BEM VINDO (A) Á DISCIPLINA: REQUISITOS DE SISTEMA**

Brooks Jr., em um antigo artigo publicado pela IEEE Computer, intitulado como No Silver Bullet: Essence and Accidents in Software Engineering (IEEE Computer, abril 1987), define: “A parte mais árdua na construção de um sistema de software é decidir o que construir. Nenhuma outra parte do trabalho compromete mais o sistema se for feito de forma imprópria. Nenhuma outra parte é mais difícil de corrigir a posteriori.”

Para os mais tecnicistas (e até muitos analistas), certamente existe uma grande dificuldade em compreender a veracidade desta mensagem. É comum nos preocuparmos bastante com a linguagem de programação, base de dados ou quaisquer outros recursos no tocante a base tecnológica. Contudo, cometemos uma grave e custosa infração: definir o que realmente o sistema deve fazer; quais são as entradas disponíveis e as saídas esperadas para o sistema.

Nesta disciplina, teremos a oportunidade de compreender a importância na definição dos requisitos de software, identificando como ela contribui para o desenvolvimento de um sistema de qualidade, ou seja, que os objetivos estão alinhados com a necessidade, evitando surpresas ou pouco entendimento sobre o que realmente se deseja. O término do estudo dessa disciplina propiciará uma visão sobre os benefícios de se conhecer bem os requisitos desejados, e como estes irão proporcionar um sucesso no que diz respeito a assertividade do produto de desenvolvimento de software, ou seja, em um sistema que realmente satisfaça as necessidades dos usuários.

### **Aula 1: Introdução a Requisitos de Sistemas**

Conceito de qualidade. Garantia de qualidade de software. Levantar e documentar bem requisitos de projetos (principalmente de projetos de software). Importância de requisitos bem especificados e documentados comentando sobre erros de sistemas, funcionalidades, que não atendem ao negócio suporte pelo software e os sistemas que sofrem manutenção por erros de especificação ou sistemas que são abandonados durante o desenvolvimento ou já depois de implementados por não atenderem de forma correta o negócio.

### **Aula 2: Níveis de Requisitos: Requisitos de Usuários e Requisitos de Sistemas.**

Identificar e diferenciar as especificidades e diferenças entre os requisitos. Os requisitos de domínio são derivados do domínio da aplicação do sistema. Os requisitos de usuário englobam funcionais e não funcionais de forma que eles sejam entendidos por usuários do sistema mesmo que não possuam conhecimento técnico detalhado. Apresentar alguns exemplos de documentação de requisitos e os problemas gerados por utilizar técnicas não específicas para a documentação, por exemplo, a linguagem natural (problemas relacionados a falta de clareza, confusão de requisitos, fusão de requisitos, erros de interpretação por usuários e desenvolvedores, etc.), exemplificar os requisitos obrigatórios e requisitos desejáveis e qual a forma correta de documentação.

### **Aula 3: Tipos de Requisitos: Funcionais, Não Funcionais e Regras de Negócios**

Apresentar os conceitos e a diferença entre os requisitos funcionais e não-funcionais. Destaque a importância das métricas para especificar os requisitos não funcionais. Conceito completeza (todos os serviços exigidos pelo usuário devem ser definidos.) e consistência (os requisitos não devem ter definições contraditórias).

#### **Aula 4: Identificação de stakeholders**

Técnicas de levantamento de requisitos. Apresentar o de stakeholders. Características dos stakeholders. Métodos para boa interação com stakeholders. Conhecer técnicas para levantamento de requisitos. Identificar as melhorias técnicas para cenários diferentes.

#### **Aula 5: Documentação de Requisitos de Software**

Capítulo introdutório. Itens do documento de requisitos: (1) os serviços e funções que o sistema deve proporcionar; (2) os constrangimentos sob os quais o sistema deve operar; (3) propriedades genéricas e constrangimentos relativos às propriedades emergentes do sistema; (4) definições de outros sistemas com os quais o sistema deverá ficar integrado; (5) informação acerca do domínio de aplicação do sistema; (6) constrangimentos do processo utilizado para desenvolver o sistema; (7) descrições do hardware no qual correrá o sistema. Utilidade de um documento de requisitos: aquisição de software pronto. Metodologias ágeis no levantamento de requisitos em diferentes abordagens no desenvolvimento de software.

#### **Aula 6: Engenharia de Requisitos e Estudos de Viabilidade**

Nessa etapa iremos imergir detalhar nos conteúdos sobre a engenharia de requisitos, inclusive nossa aula de hoje iniciar pelos fundamentos dessa área, destacando a importância no resultado de um software que atenda as necessidades dos usuários.

#### **Aula 7: Elicitação de Requisitos**

Segunda etapa das nossas aulas sobre as atividades da área de requisitos.

Continuaremos imersos nos conteúdos sobre a engenharia de requisitos; nossa aula de hoje trabalhar sobre a elicitação de requisitos, destacando a importância no resultado de um software que atenda as necessidades dos usuários.

#### **Aula 8: Validação de Requisitos**

Identificar mais uma atividade da engenharia de requisitos. Reconhecer o processo de validação de requisitos.

Verificar o motivo e a importância da validação de requisitos.

#### **Aula 9: Gerenciamento de Requisitos**

Conhecer mais uma atividade da engenharia de requisitos. Identificar a função da atividade de gerenciamento de requisitos. Reconhecer a importância do controle de mudanças.

#### **Aula 10: Casos de Uso**

Identificar a utilidade de casos de uso para a engenharia de software. Reconhecer como o caso de uso contribuiu para a área de requisitos de sistemas. Conhecer a estrutura dos casos de usos. Obter uma introdução sobre a linguagem UML.

## Bibliografia

SOMMERVILLE, I., **Engenharia de Software**, 9<sup>a</sup> Edição. Pearson, 2011.

PRESSMAN, R., **Engenharia de Software**, 6<sup>a</sup> Edição. McGraw-Hill, 2006.

NOGUEIRA, Marcelo. **Engenharia de Software**, 1<sup>a</sup> Edição. Ciência Moderna, 2009.

## Avaliação

A avaliação é continua, integradora, com ênfase nos aspectos colaborativos, incluindo tarefas coletivas, e contempla o diagnóstico, o processo e os resultados alcançados por intermédio de avaliações diagnósticas, formativas e somativas, considerando os aspectos da autoavaliação.

A avaliação somativa da aprendizagem é realizada presencialmente pelo aluno no IES e segue a normativa da Universidade. A(s) prova(s) presencial(is) segue(m) o calendário acadêmico divulgado para o aluno.

Durante o Curso, os alunos realizam atividades propostas, compostas de questões objetivas e discursivas referentes ao conteúdo estudado, podendo ser elas de autodiagnóstico ou de discussão.

Ao final dessa disciplina você será capaz de:

- Apresentar os conceitos fundamentais e importância dos requisitos no processo de desenvolvimento de software.
- Compreender o que vem a ser qualidade, e como podemos desenvolver um sistema com qualidade.
- Apresentar as atividades envolvidas para elicitação e validação dos requisitos, evidenciando a importância de cada uma delas e as diferentes técnicas utilizadas, bem como o relacionamento entre elas.
- Ter uma visão sobre a engenharia de requisitos de software.
- Desenvolver senso crítico do aluno mostrando a necessidade do gerenciamento de requisitos para apoiar cada atividade do ciclo de vida de software.

Fique atento (a) e bons estudos!