



Aula XI Desenvolvimento de SI

Eduardo Kinder Almentero

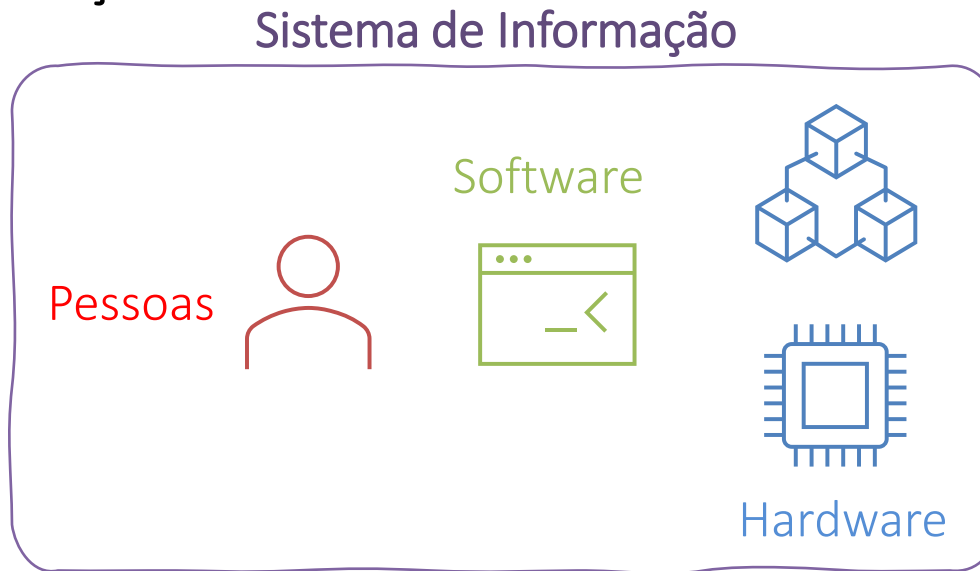
ekalmentero@gmail.com

Sumário da Aula

- Sistemas de Informação
- Definição de Software
- Guia SWEBOK
 - Engenharia de Software
 - Processo de Software
 - Engenharia de Requisitos
 - Arquitetura (Projeto)
 - Construção
 - Teste de Software
 - Implantação
 - Manutenção
- Engenharia de Software
- Processo de Software
- Economia na Engenharia de Software

Sistemas de Informação

É um conjunto de componentes interrelacionados que atuam de forma colaborativa para coletar (ou recuperar), processar, armazenar e disseminar informação, com o intuito de dar suporte a tomada de decisão em uma organização.¹



Ênfase no software?

Hardware: usualmente de propósito geral

Software: na maioria das vezes é customizado

¹Laudon, K.C., Laudon, J.P., (2016) Management Information Systems. Pearson. 13 ed

Definição de Software

- Software consiste em¹:
 1. “Instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados;”
 - Código (vocês estão vendo em LP1)
 2. “Estruturas de dados que possibilitam aos programas manipular informações adequadamente;”
 - Por exemplo, os banco de dados relacionais que vimos na apresentação do dia 23/10/2020
 3. “Informação descritiva, tanto na forma impressa como na virtual, descrevendo a operação e o uso dos programas.”
 - Documentação, incluindo evidências do que foi realizado

¹Pressman, R. S. (2011). Engenharia de Software: Uma abordagem profissional. Sétima Edição.

SWEBOK

- SWEBOK (Software Engineering Body of Knowledge)
 - Criado com o apoio da IEEE
 - IEEE e ACM são as principais instituições que promovem a disseminação de conhecimento na área.
 - Reúne **um conjunto de conhecimentos** sobre a área;
 - Tem como objetivo **promover uma visão unificada e consistente** da disciplina em todo o mundo;
 - **Não apresenta todo o conteúdo**, mas serve como um **guia** para explorar os **distintos aspectos** da disciplina.
 - Foi transformado em uma norma ISO/IEC TR 19759:2016 SWEBOK;

Engenharia de Software

- Definições IEEE:
 - Aplicação sistemática de conhecimento científico e tecnológico, métodos, e experiência ao projeto, implementação, teste e documentação de software.¹
 - Aplicação de abordagem sistemática, disciplinada e quantificável ao desenvolvimento, operação e manutenção de software, ou seja, a aplicação da engenharia ao software.²

¹ISO/IEC 2382:2015, *Information technology – vocabulary*

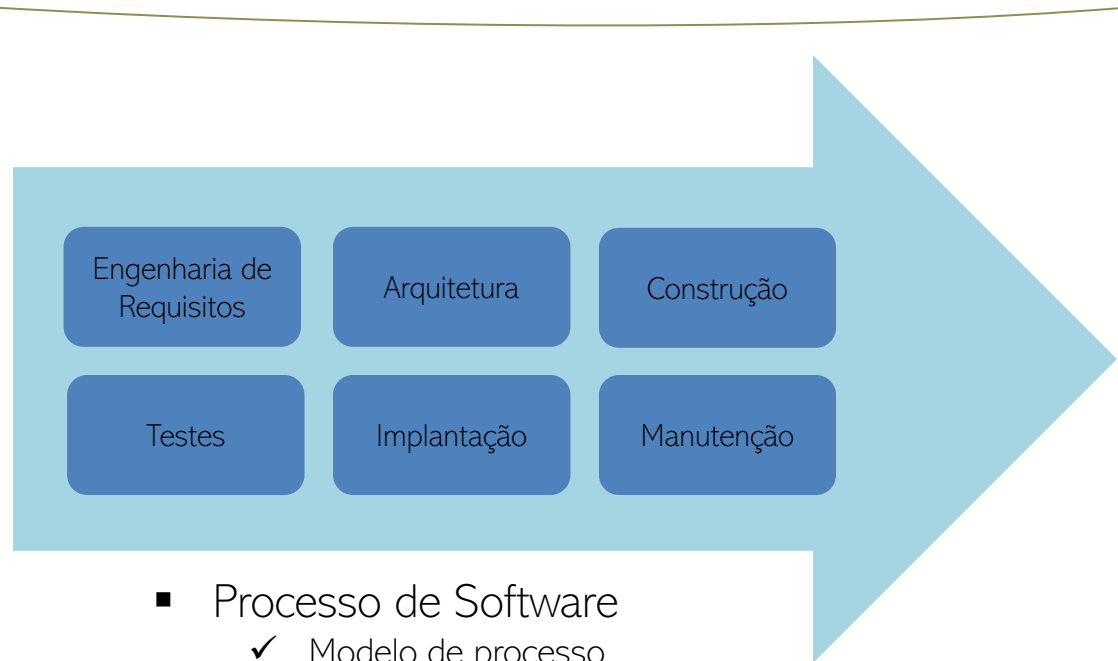
²ISO/IEC TR 19759:2016 SWEBOOK

Processo de Software

Gerenciamento de Projetos

“Mundo Real”

Problema



Software

Produto Final

- Processo de Software
 - ✓ Modelo de processo
 - ✓ Gestão de requisitos
 - ✓ Qualidade de software
 - ✓ Gerência de configuração

Processo de Engenharia de Software

- Um **processo de engenharia** consiste em um **conjunto de atividades interrelacionadas** que **transformam** uma ou mais **entradas** em **saídas**, consumindo recursos para realizar a transformação.¹
- Um **processo de engenharia de software** (ou apenas **processo de software**) consiste em **atividades realizadas** por **engenheiros de software** para **desenvolver, manter e operar software**.¹
- **Processos são especificados para:**
 - ✓ **reduzir a complexidade do trabalho** e assim **facilitar o entendimento** das pessoas, a **comunicação e coordenação**,
 - ✓ **auxiliar no gerenciamento de projetos de software**,
 - ✓ **mensurar a qualidade do software** de maneira eficiente,
 - ✓ **dar suporte a melhoria contínua** dos processos,
 - ✓ **facilitar o uso de ferramentas** a fim de dar **suporte automatizado** a execução do processo.

¹ISO/IEC TR 19759:2016 SWEBOK

Engenharia de Requisitos (ER)

- Requisitos de software expressam as **necessidades** e **restrições** a que o **software** deve **atender** afim de contribuir para a **solução de um problema** do mundo real.
 - Requisitos funcionais;
 - Requisitos não funcionais
- A ER compreende: **elicitação**, **modelagem** (especificação) e **análise**, assim como o **gerenciamento dos requisitos** durante todo o ciclo de vida do software.
 - A **elicitação** compreende a **coleta de fatos** a partir das **fontes de informação**.
 - A **modelagem**, ou **especificação**, está relacionada a **representação**, **organização** e **armazenamento** dos **fatos coletados** através do **uso de modelos**.
 - Essa representação é de suma importância para comunicação das necessidades ao longo do ciclo de desenvolvimento.
 - A **análise** inclui a **verificação** e **validação**, e tem como objetivo determinar se a descrição **está correta** e se **corresponde às necessidades reais** dos interessados.
- O papel da ER é entender e descrever o problema, e **não a solução**.

Arquitetura (Projeto)

- Atividade que dá início a construção da solução.
- Envolve o **projeto** e o **detalhamento** da arquitetura
 - O **projeto** da arquitetura é a criação de uma **estrutura e organização** de alto nível que **descreve e identifica** os vários componentes do software.
 - No **detalhamento** são descritas as **minúcias** de cada componente, possibilitando a sua posterior **construção**.
- **Todo software possui arquitetura**, ainda que não tenha sido elaborada de forma **consciente** e/ou não esteja **documentada**.
- A única forma de obter uma arquitetura que **atenda as necessidades dos interessados** e atenda os **principais padrões de qualidade** é
 - construí-la **de acordo com os requisitos** identificados e
 - adotar os **padrões e boas práticas** adequadas.
- Existem **diversas linguagens** (modelos) para descrição da arquitetura e a mesma arquitetura é **descrita através de vários modelos (visões)**.

Construção

- A **construção** (ou implementação) do **software** refere-se a **criação de software que funciona** (executa) através da **codificação, verificação, teste de unidade, teste de integração e depuração**.
- A arquitetura deve ser seguida, ou seja, seus **componentes devem ser implementados**.
 - É comum que **ajustes sejam feitos na arquitetura** durante a codificação, para contemplar questões que não foram antecipadas.
- Princípios:
 - Minimizar a complexidade;
 - Antecipar (futuras) mudanças;
 - Construção para verificação;
 - Reuso;
 - Uso de padrões.
- O seu **código não será lido/interpretado apenas pela máquina!**
 - Você e demais componentes da equipe (e outros no futuro) precisam compreendê-lo.
- Compreende, também, a escrita de **testes automatizados**
 - Requerem a escrita de código

Teste de Software

- Consiste ¹ISO/IEC TR 19759:2016 SWEBOK
 - na avaliação dinâmica do software,
 - através de um conjunto finito de casos de teste,
 - selecionados de adequadamente a partir de um conjunto geralmente infinito de possibilidades,
 - para determinar se o software possui o comportamento esperado.¹
- Testes são experimentos controlados, ineficientes, caros, ineficazes e **requerem criatividade**.
 - Ainda assim são extremamente necessários.
- Os testes só pode provar uma coisa: **que o software possui erros**.
 - Podem gerar indícios (documentação) que o software possui qualidade satisfatória.

Implantação

- **Após o teste bem sucedido** o software é colocado em produção
 - Ambiente real de operação;
 - Usuários reais;
- Pode ser necessário:
 - Treinar usuários;
 - Configurar o ambiente de produção;
 - Converter bases de dados;
 - Desativar softwares que serão substituídos;
 - etc
- Nesta etapa são realizados **testes de aceitação**
 - Buscam a **aprovação formal e final** do cliente a partir de **critérios pré-determinados**.
- Se o **software satisfizer as condições determinadas** no teste de aceitação **ele é aceito** e sua operação iniciada.

Manutenção

- O resultado do processo de software é um **produto** que **satisfaz as necessidades** do usuário.
- Para se **manter útil**, com o passar do tempo o software **precisa ser modificado**
 - Quando os software está em produção:
 - erros são descobertos,
 - sistemas operacionais são modificados,
 - novas necessidades dos usuários surgem.
- A **manutenção compreende** desde uma **alteração pontual**, até uma **modificação completa**, que pode ser feita através de um **novo processo de software**.
- Costuma ser a **fase mais cara (e longa)** do ciclo de vida do **software**, pois se **estende-se por toda sua vida útil**.

Gerência de Configuração

- Um sistema pode ser visto como uma coleção de versões específicas de itens de hardware e software, combinados através de procedimentos específicos para servir a um propósito em particular.
- Ao longo do ciclo de vida do software, incluindo seu desenvolvimento e manutenção, estes itens são continuamente modificados, dando origem a várias versões.
- A gerência de configuração é “a disciplina de identificar a configuração de um sistema em pontos distintos no tempo com o objetivo de controlar sistematicamente as mudanças na configuração e manter a integridade e rastreabilidade da configuração ao longo do ciclo de vida do sistema”¹

¹ISO/IEC TR 19759:2016 SWEBOK

Qualidade de Software

- Está relacionada a capacidade do software de **satisfazer as necessidades explícitas e implícitas** quando utilizado sob condições especificadas (ISO/IEC 25000 – SQuaRE)
- “Conformidade aos **requisitos de desempenho e de funcionalidade que foram explicitamente definidos**, aos **padrões de desenvolvimento explicitamente documentados** e **às características implícitas** que são esperadas por todo software desenvolvido por profissionais”¹
- Existem diversos **modelos, padrões e normas** associados a qualidade de software
 - CMMI, MPS.BR, ISO/IEC 25000 etc

¹Pressman, R. S. (2011). Engenharia de Software: Uma abordagem profissional. Sétima Edição.

Economia na Engenharia de Software

- Quando consideramos a **engenharia de software** no **contexto organizacional** aspectos como **valor, custos e uso de recursos** são **críticos**.
- Uma decisão sobre o **uso de uma tecnologia** pode ser **trivial do ponto de vista técnico**, mas **complexa**, com **implicações sérias** na **viabilidade do negócio**.
- A **economia** na engenharia de software é a **disciplina** que tem com objetivo alinhar as decisões relacionadas ao software com os **interesses do negócio**.



FIM

Eduardo Kinder Almentero

ekalmentero@gmail.com