

AULA 1 – INTRODUÇÃO À ENGENHARIA DE SOFTWARE

OBJETIVO DA AULA

Compreender os conceitos iniciais sobre software e a aplicação do contexto de engenharia ao cenário de produção de software.

APRESENTAÇÃO

O profissional de tecnologia da informação é responsável por uma atividade essencial no dia a dia de qualquer negócio: a resolução de problemas.

Os softwares são programas de computadores desenvolvidos para um cliente específico ou para o mercado em geral, e visam possibilitar com que os seus usuários tenham o seu trabalho afetado positivamente por meio da tecnologia.

O contexto de engenharia atribuído à elaboração de software tem o objetivo de aplicar métodos, práticas e ferramentas na resolução de problemas, visando garantir eficiência, praticidade e segurança aos processos apoiados por ele.

Nesta aula vamos aprender um pouco sobre o papel da engenharia de software na elaboração de softwares.

1. A IMPORTÂNCIA DO SOFTWARE NA ATUALIDADE

Hoje é praticamente impossível pensar em qualquer atividade pessoal ou profissional sem o apoio de um software. Os softwares podem estar presentes em pequenas soluções através de sistemas embutidos, em geladeiras e TVs, por exemplo, ou grandes sistemas de informação, como os de bancos e companhias aéreas.

A nossa vida pessoal também é muito afetada diariamente por conta do uso dos softwares, e neste caso podemos citar o controle dos nossos compromissos diários em agendas eletrônicas, marcações de consultas médicas, até a compra de ingressos para shows e cinema, isso sem contar com todas as facilidades que os softwares que estão nos smartphones nos proporcionam.

Vale a pena dizer que o profissional de tecnologia da informação, por ter como especialidade a resolução de problemas, acaba transitando pelos diversos níveis organizacionais, a saber, operacional, tático e estratégico.

Livro Eletrônico

FIGURA 1 | **Pirâmide Organizacional**



Fonte: Laudon e Laudon.

Para cada nível, necessidades diferentes são observadas, pois o gerente do nível estratégico tem a necessidade de observar o dado de uma maneira diferente da visão que necessita o gerente operacional.

Assim, toda vez que o profissional de software estiver envolvido na produção de algum software, é importante conhecer bem os *stakeholders*. Os *stakeholders* são os profissionais interessados e envolvidos no projeto, aqueles que podem tomar decisão durante o andamento do projeto.

Alguns exemplos de *stakeholders* são os gerentes, o patrocinador do projeto e os *key-users*, que são usuários importantes da parte do cliente. É importante ressaltar que alguns dos próprios membros da equipe de desenvolvimento também são considerados *stakeholders*.

2. SOFTWARE

Mas o que é o software afinal? Em um contexto simples, os softwares são programas de computadores criados para a resolução de um ou mais problemas.

Esses programas podem ser desenvolvidos como produtos que vão apoiar o trabalho de um ou mais profissionais. Esses produtos podem ser divididos em duas categorias: os softwares genéricos e os softwares por encomenda.

Os softwares genéricos são desenvolvidos para que sirvam para várias empresas de um mesmo ramo. Por exemplo, um software para controlar o estoque de uma empresa pode ser facilmente empregado em outra empresa que também deseja controlar o seu estoque.

Já os softwares por encomenda tratam especificamente o problema do cliente solicitante, sendo desenvolvido para uma realidade particular e, portanto, pertencente exclusivamente a quem o encomendou. Um exemplo desse tipo de software seria um sistema construído para controlar as atividades acadêmicas de uma determinada escola.

3. O PROFISSIONAL DE SOFTWARE

A partir daí temos o profissional de software atuando como facilitador técnico para que outros profissionais possam ter as suas atividades diárias influenciadas pelo apoio dos softwares.

A tarefa desse profissional é observar e entender as necessidades do cliente, para poder analisar e propor uma solução computacional que tornará viável, ágil e simplificada a execução das tarefas do cliente.

Este é um ponto interessante para o profissional de tecnologia da informação, de uma forma geral, pois ele passa a conhecer o domínio de negócio ao qual ele está prestes a construir uma solução, entendendo o funcionamento, os processos e as regras de negócio. Uma responsabilidade e tanto, não acha?

A área de tecnologia proporciona diversos desafios que devem ser encarados pelo profissional de TI. Cada um desses desafios está relacionado com o papel e cargo desses profissionais. Por ser uma tarefa muito difícil elencar todos os papéis que um profissional de TI pode possuir, alguns dos mais comuns conseguimos observar na lista a seguir:

a) Desenvolvimento: apoiam atividades organizacionais a partir de sistemas que facilitam o controle, coordenação e tomada de decisão nos negócios.

- Analista de Sistemas;
- Analista de Requisitos;
- Programador;
- Analista de Testes;
- Administrador de Dados.

b) Suporte: tem a responsabilidade de manter os sistemas computacionais e a infraestrutura em funcionamento e com a garantia da segurança.

- Analista de Suporte;
- Analista de Redes;
- Analista de Infraestrutura;
- Administrador de Banco de Dados.

Além dessas duas grandes áreas, ainda temos cargos relacionados à Análise dos Dados, Governança, dentre outras áreas relacionadas à tecnologia da informação.

4. ENGENHARIA DE SOFTWARE

Para que tudo isso seja possível, nasceu a Engenharia de Software, uma disciplina que prevê o desenvolvimento profissional de software, propondo métodos, técnicas e ferramentas que permita o desenvolvimento de um produto com qualidade.

Muito antes de tratar o desenvolvimento de software como engenharia, ocorreu a chamada de “Crise do Software”. Esse termo se origina dos anos 1970 e relata o momento em que a Engenharia de Software praticamente não existia, algo que se tornou crítico, uma vez que esse período apresentou um crescimento substancial na atividade de desenvolvimento de software em todo o planeta.

Especificamente este termo foi cunhado na apresentação de uma palestra feita por Edsger Dijkstra na *Association for Computing Machinery The Humble Programmer* (EWD340), que gerou artigo publicado no periódico *Communications of the ACM*.

O artigo *The Humble Programmer* (1972) pode ser encontrado no link www.cs.utexas.edu/users/EWD/ewd03xx/EWD340.PDF.

Alguns aspectos conseguem direcionar quais eram os sintomas que apontavam a Crise do Software:

- Projetos que não cumpriam o orçamento previsto;
- Projetos que ultrapassavam o prazo determinado no início do desenvolvimento;
- Software com defeito;
- Software com qualidade pobre e muito difíceis de serem entendidos;
- Projetos difíceis de manter;
- Softwares que não entregavam aquilo que era pedido pelo cliente.

Nesse aspecto, a Engenharia proporcionou aos profissionais de desenvolvimento uma espécie de ponto zero, que ajudava os desenvolvedores a utilizarem e compartilharem as melhores práticas para o desenvolvimento, sempre a partir da definição de técnicas, ferramentas e processos que foram bem-sucedidos em projetos diversos.

5. CAUSAS DA FALHA DE PROJETOS DE SOFTWARES

Ainda com todas as definições da Engenharia de Software, os projetos podem falhar caso não sejam considerados alguns aspectos importantes. A lista a seguir apresenta alguns dos motivos clássicos que fazem com que os projetos acabem falhando:

- Escolha errada da tecnologia a ser empregado no projeto;
- Escolha errada do processo de desenvolvimento do software;

- Mudanças rápidas nas regras de negócio;
- Inflexibilidade do sistema ou do processo;
- Pouco tempo de investimento no planejamento do projeto;
- Prazos irreais (subestimação ou superestimação do esforço);
- Falta de recursos adequados (pessoas, máquinas, ferramentas, ambiente etc.);
- Falhas no processo de captura, especificação e gerência de requisitos;
- Baixa participação dos *stakeholders* no projeto;
- Baixo índice de *feedback* pelo cliente;
- Falhas no gerenciamento de riscos;
- Pouco suporte da alta direção.

O profissional de software deve estar atento para cada uma dessas causas e deve considerar os possíveis planos de contingência para o caso de falhas nos quesitos mais importantes, principalmente aqueles que se tem menor controle, como a baixa participação dos *stakeholders*.

6. OS DESDOBRAMENTOS DA ENGENHARIA DE SOFTWARE

O desenvolvimento de software passou por uma evolução natural agregando valores entre pessoas, hardware e softwares disponíveis.

Antes, os softwares eram produzidos para atender a problemas departamentais. Por exemplo, softwares de contabilidade, departamento de pessoal e estoque. Mais precisamente, o objetivo era a automação de processos.

Mas as empresas entenderam que era necessário que os softwares de cada parte de uma organização se comunicassem, evitando assim o registro de informações em duplicidade. Agora o foco estava no aumento da produtividade.

A integração dos sistemas foi essencial para que as informações pudessem ser interpretadas de maneira corporativa, gerando eficiência e criando um diferencial competitivo em relação aos concorrentes diretos.

A aplicação de metodologias de desenvolvimento e o uso de ferramentas que apoiam esses processos tornou-se uma prática, aumentando a qualidade e permitindo que todo um processo de continuidade e aperfeiçoamento do software se tornasse uma realidade.

Atualmente outras iniciativas para o aumento na produtividade do desenvolvimento do software são aplicadas nas fábricas de software, como o compartilhamento de projetos construídos em equipe, o uso de ferramentas de versionamento, e a aplicação dos modelos ágeis de desenvolvimento, fazendo com que a elaboração de software seja facilitada, e que as melhores práticas sejam reconhecidas e aplicadas, tudo isso graças à Engenharia de Software.

CONSIDERAÇÕES FINAIS

Nesta aula inicial abordamos não só a conceituação de software, mas também dedicamos um tempo a entender o importante papel do analista de sistemas nos processos que apoiam a construção de soluções para os mais diversos problemas dos mais distintos domínios.

Traçamos um panorama sobre a forma como o software é importante em nosso dia a dia atualmente, e como era o cenário descrito como “crise do software” em tempos em que não havia a preocupação com a metodologia e os processos para desenvolvimento de software, assim como havia poucas ferramentas de apoio para a atividade de desenvolvimento de software.

Descrevemos também a evolução no conceito de produção de software, apontando quais foram as principais evoluções e como estas ajudaram na melhoria desta atividade essencial para os dias atuais.

MATERIAIS COMPLEMENTARES

Artigo que deu origem ao termo “crise do software” de Edsger Dijkstra: *The Humble Programmer*. Disponível em: <https://www.cs.utexas.edu/users/EWD/ewd03xx/EWD340.PDF>. Acesso em: 28 out. 2022.

Artigo: Por que os Softwares falham? Disponível em: <http://www.linhadecodigo.com.br/artigo/2054/por-que-projetos-de-software-falham.aspx>. Acesso em: 28 out. 2022.

REFERÊNCIAS

KENNETH C. LAUDON; JANE P. LAUDON. *Sistemas de informação gerenciais*. 11ª edição. Editora Pearson, 2014.

PRESSMAN, R. G. *Engenharia de Software*. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. *Engenharia de Software*. 10ª ed. São Paulo: Pearson Addison. Wesley, 2019.