

TEXTO DE APOIO



AULA 1

Introdução a Engenharia de Software

Professora Kassya Christina Rigolon de Andrade



Universidade Presbiteriana
Mackenzie





Sumário



INTRODUÇÃO	3
INTRODUÇÃO À ENGENHARIA DE SOFTWARE.....	4
SOFTWARE E ENGENHARIA DE SOFTWARE	5
REFERÊNCIAS	15

INTRODUÇÃO

Desenvolver softwares é uma atividade muito complexa. A Engenharia de Software é uma área da ciência da computação que busca apresentar as melhores técnicas para especificação, criação, desenvolvimento e manutenção de sistemas de software, com aplicação de tecnologias, práticas de gerência de projetos e outros campos do conhecimento, visando à organização, à produtividade e à qualidade.

Portanto, a disciplina Introdução à Engenharia de Software tem como objetivo maior dar uma visão geral sobre todas as fases que envolvem o processo de desenvolvimento de um software e apresentar algumas ferramentas utilizadas em tais fases.

Basicamente, os assuntos tratados nesta disciplina são: “Introdução à Engenharia de Software”, “Introdução à Engenharia de Requisitos e Técnicas da Engenharia de Requisitos”, “Projeto e Implementação” e, finalmente, “Qualidade, Testes e Evolução de Software”.

Nas Aulas 1 e 2, conheceremos o principal objetivo da engenharia de software e os problemas enfrentados durante seu desenvolvimento; seremos apresentados a alguns paradigmas e modelos de processos envolvidos nessa área, os quais podem auxiliar na minimização desses desafios.

Nas Aulas 3 e 4, estudaremos os princípios fundamentais da Engenharia de Requisitos, que é a parte mais importante do processo de desenvolvimento de software, na qual se identificam as necessidades ou os requisitos de um cliente. Além disso, trabalharemos também com algumas técnicas e ferramentas utilizadas para obter e registrar os requisitos do cliente.

Nas Aulas 5 e 6, veremos os conceitos de arquitetura de software e a importância do projeto de arquitetura, como também analisaremos questões chaves que devem ser consideradas na implementação de um software, que é a fase seguinte após o projeto no processo de seu desenvolvimento.

Nas Aulas 7 e 8, conheceremos os estágios de teste de software, desde o desenvolvimento até a aceitação do software pelos clientes. Descobriremos também o porquê de a evolução do software ser uma parte tão importante da engenharia de software.

Sendo assim, desejo um excelente curso a você!

INTRODUÇÃO À ENGENHARIA DE SOFTWARE

Olá! Nesta primeira aula, estudaremos o que é Engenharia de Software, trabalhando com sua importância dentro da área da computação, entendendo que existem diferentes tipos de sistemas de software e que cada um exige técnicas diferentes de engenharia, assim como observaremos questões éticas e profissionais importantes para os engenheiros de software.

O desenvolvimento e a criação de um software são atividades muito complexas. Há anos, quando os computadores surgiram, a construção de software era realizada de forma bem informal, porque não havia, ainda, técnicas ou práticas desenvolvidas, por falta de experiência em construção de software. Com o passar do tempo, os softwares ficaram mais complexos, e os problemas com prazo de entrega, custo e qualidade no desenvolvimento aumentaram.

Percebeu-se, então, que a experiência inicial e o processo informal de desenvolvimento de software não eram suficientes para a entrega de um produto com êxito. Novas práticas eram necessárias para controlar todo o processo de desenvolvimento de software, principalmente o dos mais complexos. É justamente isso que propõe a Engenharia de Software. Hoje, existem diversas técnicas e métodos no mercado de trabalho para desenvolver softwares com custo e qualidade adequados às necessidades do cliente e com entrega dentro do prazo previsto.

Sendo assim, o que é um método ou uma técnica?

Um método ou técnica é um procedimento formal para produzir algum resultado. Por exemplo, um chefe de cozinha pode preparar um molho empregando ingredientes sequencialmente combinados em uma ordem e momento específicos, de tal maneira que o molho engrosse, mas também não coagule e nem desande. O procedimento para preparar o molho envolve preocupação com o tempo e os ingredientes, mas pode não depender do tipo de equipamento de cozinha utilizado. (PFLEEGER, 2004, p. 3).



Seria possível comparar a construção de um software com a de uma casa? Há diferenças? Quais seriam? Liste o passo a passo da construção de uma casa e, depois da Aula 2, você poderá observar com mais entendimento se a comparação é possível.

Nosso desafio, nesta primeira aula, será compreender o que é Engenharia de Software e qual é sua importância na área da computação.

SOFTWARE E ENGENHARIA DE SOFTWARE

Grande parte da população, hoje em dia, é dependente de software, ou seja, de sistemas baseados em computadores. Para desenvolver um software com qualidade, baixo custo e prazos adequados, os desenvolvedores precisam dominar os métodos e as técnicas apresentados na Engenharia de Software. Antes de conhecermos tais métodos e técnicas, precisamos entender alguns conceitos.

Então vamos lá!

O que é Software? Softwares são todos os programas de computador, assim como sua documentação e configuração, que, associados, cooperam para que o programa funcione corretamente. Os produtos de software podem ser de dois tipos:

- Genéricos: desenvolvidos para serem vendidos a uma grande variedade de clientes (por exemplo: Excel, Word, PowerPoint e aplicativos para dispositivos móveis).
- Personalizados: desenvolvidos para um único cliente de acordo com suas especificações (por exemplo, Enterprise Resource Planning – ERP, sistemas de controle de dispositivos eletrônicos, sistema de controle de tráfego aéreo).

Saiba mais!

Na área de Tecnologia da Informação (TI), ouve-se muito falar em sistema SAP, que é um tipo de ERP. Hoje em dia, profissionais com experiência em SAP são muito requisitados no mercado. SAP SE é uma empresa de origem alemã, criadora de software de gestão de empresas (ERP). Ao longo de quatro décadas, a SAP evoluiu de uma empresa pequena e regional na cidade de Waldorf para uma organização de alcance mundial. Sugerimos visitar o site da empresa para saber mais sobre a SAP e para conhecer melhor o que é um ERP.



O que é Engenharia de Software (ES)? É uma disciplina relacionada a todos os aspectos da produção de um software. Assim como a Engenharia Civil é o ramo da engenharia que projeta e executa obras como edifícios, pontes, viadutos, estrada e barragens, a ES propõe ferramentas, técnicas e processos para:

- entender com precisão qual é o problema, ou seja, quais são as necessidades associadas ao sistema que deve ser construído/modificado;
- produzir uma solução adequada para esse problema, ou seja, um sistema pronto para ser usado, levando em consideração as necessidades das partes interessadas (usuários);
- levar em conta restrições de desenvolvimento e recursos disponíveis.

A Figura 1 mostra as camadas da engenharia de software que possibilitam o desenvolvimento de software de forma racional e dentro do prazo, sempre focando na qualidade.

Figura 1– Camadas da Engenharia de Software



Fonte: Pressman (2011, p. 16).

Descrição da imagem: ilustração apresentando quatro camadas da engenharia de software com as palavras Ferramentas, Métodos, Processo e Foco na qualidade.

Qual é a diferença entre engenharia de software e ciência da computação?

A ciência da computação está relacionada às teorias e aos métodos que constituem a base de computadores e sistemas de software. A engenharia de software se dedica a tentar minimizar os problemas práticos da produção de software.

Qual é a diferença entre engenharia de software e engenharia de sistemas?

A engenharia de sistemas é uma disciplina mais antiga que a engenharia de software e está mais focada no desenvolvimento e na organização de sistemas tecnológicos complexos – que integram outros sistemas e envolvem grande quantidade de variáveis, como fabricantes de aviões, automóveis, navios e equipamentos de grande porte, como plataformas marítimas de petróleo e infraestrutura de telecomunicações. As técnicas de engenharia de software são usadas nos processos de engenharia de sistemas.

O que é um processo? Segue um conceito de processo, segundo (PFLEEGER, 2004, p. 36):

Quando fornecemos um serviço ou criamos um produto, seja desenvolvendo um software, escrevendo um relatório ou fazendo uma viagem de negócios, sempre seguimos uma sequência de etapas para completar um conjunto de tarefas. As tarefas são, geralmente, realizadas na mesma ordem todas as vezes: por exemplo, você não reboca uma parede sem antes colocar a tubulação, nem assa um bolo sem antes misturar os ingredientes. Podemos considerar um conjunto de tarefas ordenadas como sendo um processo: uma série de etapas que envolvem atividades, restrições e recursos para alcanças a saída desejada.

Um processo geralmente envolve um conjunto de ferramentas e técnicas.

O que é um processo de software? É um conjunto estruturado de atividades, práticas, ferramentas e artefatos necessários para o desenvolvimento de um sistema de software. São quatro as atividades fundamentais de um processo de software: especificação, desenvolvimento, validação ou teste e evolução. Uma atividade adere a um ou mais modelos de processo que veremos mais adiante.

Diferentes tipos de sistemas necessitam de diferentes processos de desenvolvimento? Sim, pois não há métodos universais de engenharia de software que sejam adequados para todos os tipos de sistemas e empresas, e o conjunto de métodos e ferramentas de engenharia de software evoluiu e evolui junto com os diferentes tipos de sistemas que vão surgindo. Os métodos e as ferramentas que a engenharia de software proporciona devem ser aplicados de acordo com o tipo de sistema que é apresentado. Então, vamos conhecer alguns tipos diferentes de sistemas de software:

- Aplicações stand-alone: são executadas em um computador pessoal ou em aplicativos que rodam em dispositivos móveis. Incluem toda a funcionalidade necessária e podem não necessitar de uma conexão com uma rede. Exemplos: programas CAD, software de manipulação de imagens, aplicações

de escritório em um computador pessoal.

- Aplicações interativas baseadas em transações: são executadas em um computador remoto e são acessadas por diversos usuários a partir de seus próprios computadores ou dispositivos móveis.
- Sistemas embarcados: controlam e gerenciam dispositivos de hardware. Exemplos: software de um celular, software que controla freio ABS no automóvel, software em um forno micro-ondas.
- Sistemas de processamento em lotes (batch): são sistemas de negócios que processam dados em grandes lotes. Exemplos: contas de telefone, sistema de folha de pagamento, sistema de restituição de imposto de renda.
- Sistemas de entretenimento: são jogos de gêneros variados que podem ser executados em um console concebido especificamente para essa finalidade. Exemplo: jogos para Playstation 4.
- Sistema para modelagem e simulação: são desenvolvidos por cientistas e engenheiros para modelar processos físicos.
- Sistemas de coleta de dados e análise: fazem a coleta de dados em um ambiente e enviam esses dados a outros sistemas para processamento.
- Sistemas de sistemas: são utilizados em organizações e são compostos de uma série de outros sistemas de software. Exemplo: ERP.
- Sistemas que são ambientes de suporte: são um conjunto integrado de ferramentas de software, utilizados para dar suporte a algum tipo de atividade. Exemplos: ambientes de programação, como Eclipse; ambiente digital de aprendizagem, como o Moodle.
- Sistemas de Informação: são utilizados para gerenciar e proporcionar acesso a um banco de dados de informações. Exemplo: sistema de registro vendas de uma loja, sistema de agendamento em um consultório médico.

O que é um modelo de processo de software? É uma representação simplificada das atividades que constituem o processo de software, seus produtos e os papéis das pessoas envolvidas na engenharia. Alguns exemplos dos tipos de modelos gerais de processo de software que podem ser produzidos são:

- Modelo de workflow: apresenta uma sequência de atividades dentro de um

processo.

- Modelo de fluxo de dados: apresenta o fluxo das informações dentro de um processo.
- Modelo de papel/ação: apresenta os papéis e as funções realizadas pelas pessoas envolvidas no processo.

A maioria dos modelos de processos de software é baseada em um dos três modelos gerais ou nos paradigmas de desenvolvimento de software: o modelo em cascata, o desenvolvimento iterativo e a engenharia baseada em componentes. Cada um desses modelos será estudado com mais detalhes adiante.

Existe um modelo de processo de software ideal? Não existe um processo ideal, ele é adaptável de acordo com o contexto, podendo ser mais formal ou mais flexível. Processos de software são complexos e dependem de vários fatores como:

- produto a ser desenvolvido;
- equipe;
- recursos disponíveis.

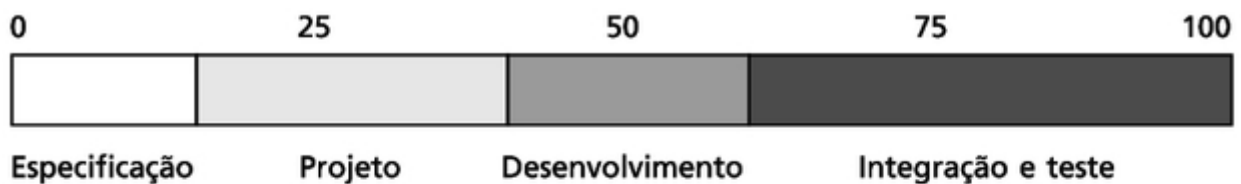
O que são artefatos de software? Artefatos de software são os diferentes tipos de subprodutos concretos produzidos durante o desenvolvimento de software. Exemplos: diagramas, documentos de requisitos do projeto que descrevem a função, arquitetura e design do software, planos de projetos, processos de negócios, avaliações de riscos, manuais, arquivos executáveis, módulos, bibliotecas e outros.

Quais são os custos da engenharia de software? Não há uma resposta simples para essa pergunta, pois o custo depende do modelo de processo (que conheceremos na Aula 2) e do tipo de software que está sendo desenvolvido.

O gráfico 1 apresenta os custos de cada fase no desenvolvimento de software para cada tipo de processo (Modelo cascata, Desenvolvimento Iterativo e Baseado em Componentes) e, por final, uma visão geral dos custos nas fases de desenvolvimento e evolução.

Gráfico 1 – Distribuição dos custos nas atividades da engenharia

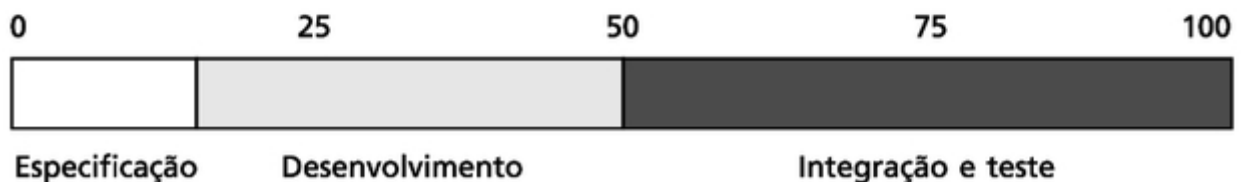
Modelo cascata



Desenvolvimento iterativo



Engenharia de software baseada em componentes



Fonte: Sommerville (2007, p. 8).

Descrição do gráfico: apresentação da distribuição dos custos nas atividades de engenharia. No primeiro item, Modelo Cascata, há uma maior concentração de custos na fase de Integração e Teste, seguido das fases de Projeto, Desenvolvimento e Especificação. No segundo item, Desenvolvimento Iterativo, há uma maior concentração de custos na fase de Desenvolvimento Iterativo, seguido das fases de Teste de Sistema e Especificação. No terceiro item, Engenharia de software baseada em componentes, há uma maior concentração na fase de Integração e Teste, seguido das fases de Desenvolvimento e Especificação. Por fim, apresenta-se os Custos de desenvolvimento e evolução ao longo da vida do software, em que a maior concentração de custos está na fase de Evolução do sistema, seguida da fase de Desenvolvimento de sistema.

O que são ferramentas CASE? Computer-Aided Software Engineering (CASE) são ferramentas ou sistemas que apoiam as atividades de desenvolvimento de software. Normalmente, apoiam um método específico. São divididas em dois tipos:

- Upper-CASE (CASE de alto nível): ferramentas para apoiar as atividades iniciais de processo de requisitos e de projeto.
- Lower-CASE (CASE de baixo nível): ferramentas para apoiar as atividades finais tais como programação, debug e teste.

Quais são os atributos de um bom software?

- Funcionalidade: deve funcionar como solicitado pelo cliente.
- Desempenho: deve funcionar dentro do tempo exigido pelos usuários.
- Facilidade de manutenção: deve ser fácil de ajustar, e sua evolução deve exigir baixo custo.
- Confiabilidade: deve funcionar conforme a especificação do cliente.
- Eficiência: deve usar o mínimo de recursos e prover a máxima funcionalidade.
- Usabilidade: deve ser compreensível e de fácil uso.
- Segurança: deve ser protegido para que usuários maliciosos não consigam acessar ou danificar o sistema.

A Engenharia de Software é diferente para software web? As ideias fundamentais de engenharia de software aplicam-se ao software web da mesma forma que a outros tipos de software, mas alguns pontos importantes devem ser levados em consideração:

- O reuso de software se tornou uma abordagem dominante para construir sistemas web.
- Os sistemas web são desenvolvidos e entregues de modo incremental, isto é, as funcionalidades vão sendo disponibilizadas aos poucos. Dessa forma, não é possível especificar todos os requisitos antecipadamente.
- O software pode ser implementado usando engenharia orientada a serviços (SOA, do inglês service-oriented architecture).

- Foram criadas tecnologias de desenvolvimento de interfaces para dar suporte à criação de interfaces dinâmicas dentro de um navegador web como: AJAX e HTML5.

Quais são as responsabilidades profissionais e éticas dos engenheiros de software? Os engenheiros de software devem se comportar de modo honesto e eticamente responsável para serem respeitados como profissionais. A seguir estão listadas algumas preocupações.

- **Confidencialidade:** os engenheiros de software devem respeitar a confidencialidade de seus funcionários e clientes, independentemente de terem assinado um acordo formal ou não. Caso não aceitem essas condições, devem deixar isso explícito para seus contratantes.
- **Competência:** os engenheiros não devem conscientemente aceitar um trabalho que esteja fora de sua competência.
- **Direitos sobre propriedade intelectual:** os desenvolvedores devem ter ciência das leis locais que regem o uso de propriedade intelectual, tais como patentes, direitos autorais, transferência de tecnologia e know-how etc.; devem também tomar cuidado para assegurar que a propriedade intelectual dos funcionários e clientes seja protegida.
- **Mau uso do computador:** se aproveitar de habilidades técnicas para usar indevidamente os computadores de outras pessoas é agir de forma desonesta que comprometerá a reputação do profissional.

Curiosidade

Sociedades e instituições profissionais têm um importante papel a desempenhar, definindo padrões de ética. Organizações como Association for Computer Machinery (ACM), Institute of Electrical and Electronic Engineers (IEEE) e British Computer Society publicaram um código de conduta profissional (ou código de ética). “Computer society and ACM approve software engineering code of ethics” (ROGERSON; MILLER; GOTTERBARN, 1999). Leia, a seguir, o Código de Ética da ACMM/IEEE-CS:

Código de Ética e Prática Profissional da Engenharia de Software

Força-tarefa conjunta da ACM/IEEE-CS para ética e práticas profissionais da engenharia de software

Prefácio

A versão reduzida do código resume as aspirações em um alto nível de abstração; as cláusulas incluídas na versão completa fornecem exemplos e detalhes de como essas aspirações mudam o nosso modo de agir como profissionais de engenharia de software. Sem as aspirações, os detalhes podem se tornar legalistas e tediosos; sem os detalhes, as aspirações podem ficar pomposas, porém vazias; juntos, as aspirações e os detalhes formam um código coeso.

Os engenheiros de software devem se comprometer a fazer da análise, especificação, projeto, desenvolvimento, teste e manutenção do software uma profissão útil e respeitada. De acordo com o seu compromisso com a saúde, segurança e bem-estar do público, os engenheiros de software devem obedecer aos oito princípios a seguir:

1. Público – Os engenheiros de software devem agir coerentemente com o interesse público.

2. Cliente e empregador – Os engenheiros de software devem agir de uma maneira que atenda aos interesses de seu cliente e empregador, coerente com o interesse público.

3. Produto – Os engenheiros de software devem assegurar que seus produtos e modificações relacionadas cumpram o máximo possível os mais altos padrões profissionais.

4. Opinião – Os engenheiros de software devem manter a integridade e independência em sua opinião profissional.

5. Gestão – Os gestores e líderes em engenharia de software devem aceitar e promover uma abordagem ética do gerenciamento do desenvolvimento e manutenção do software.



6. Profissão – Os engenheiros de software devem promover a integridade e reputação da profissão em conformidade com o interesse público.

7. Colegas – Os engenheiros de software devem ser justos e apoiar os colegas.

8. Caráter – Os engenheiros de software devem aderir a uma aprendizagem contínua durante toda a vida no que diz respeito à prática de sua profissão e promover uma abordagem ética para a prática da profissão.

Fonte: SOMMERVILLE (2019, p. 15).

São estes, então, os principais conceitos e entendimentos que precisamos ter para começar a conhecer tudo o que envolve a Engenharia de Software; só com essa introdução, podemos ter uma noção da importância da Engenharia de Software, pois o mundo moderno não funciona sem software.

REFERÊNCIAS

GOTTERBARN, Don; MILLER, Keith; ROGERSON, Simon. Software Engineering Code of Ethics is Approved. 1999. Disponível em: <https://www.researchgate.net/publication/220424674_Software_Engineering_Code_of_Ethics_is_Approved>. Acesso em: ago. 2020.

PFLEEGER, S. L. Engenharia de software: teoria e prática. 2. ed. São Paulo: Pearson Prentice Hall, 2004.

PRESSMAN, R. Engenharia de Software: uma abordagem profissional. 9. ed. São Paulo: Bookman, McGraw-Hill, 2011.

SOMMERVILLE, I. Engenharia de software. 8. ed. São Paulo: Pearson Prentice Hall, 2007.

SOMMERVILLE, I. Engenharia de software. 10.ed. São Paulo: Pearson Prentice Hall, 2019.