A Engenharia de Software e os Processos de Desenvolvimento: Fundamentos e Atividades Essenciais

**Autoria:** Baseado nos Materiais de Claudia Werlich, Regina Fedozzi, Samuel Gonçalves da Silva, Rui Lopes e Diretrizes do Inep

**Resumo:** Este artigo apresenta uma visão introdutória sobre os fundamentos da engenharia de software e a estrutura dos processos de desenvolvimento. Aborda a natureza e evolução do software, a importância da engenharia de software, o papel do analista de sistemas e as atividades genéricas de um processo de software. Discute também os diferentes fluxos de processo e as abordagens para avaliação e melhoria contínua, visando a produção de software de qualidade.

**Palavras-chave:** Engenharia de Software; Processo de Software; Requisitos; Analista de Sistemas; Desenvolvimento de Software.

--------------------------------------------------------------------------------

**1. Introdução**

A tecnologia está cada vez mais integrada ao nosso estilo de vida, com inovações como computação em nuvem, realidades virtual e aumentada, e reconhecimento facial. Nesse cenário, o **software** surge como uma inovação crucial, proporcionando eficiência e agilidade quando combinado com o hardware. A crescente demanda por softwares eficientes e inovadores impulsiona a necessidade de mão de obra especializada e de técnicas eficazes para a criação de novos sistemas, o que destaca a importância da **Análise e Modelagem de Sistemas**.

A Unidade 1 do livro "Análise e Modelagem de Sistemas" explora a evolução do desenvolvimento de software, a relevância da engenharia de software para aprimorar os processos, e o desenvolvimento ágil. O objetivo principal é capacitar o leitor a reconhecer e identificar os processos de software e suas metodologias, além de compreender o papel do analista de sistemas. Este estudo visa aprofundar a compreensão desses fundamentos, cruciais para a produção de sistemas computacionais de informação de qualidade.

**2. Fundamentos da Engenharia de Software**

A engenharia de software é um campo dinâmico e desafiador, dada a singularidade de cada sistema e a constante evolução tecnológica. Ela busca fornecer disciplina, adaptabilidade e agilidade no desenvolvimento.

**2.1. A Natureza e Evolução do Software**

Software, de acordo com Sommerville (2011, p. 4), consiste em "programas de computadores com uma documentação associada", enquanto Pressman (2016) o descreve como um produto desenvolvido e suportado por profissionais de TI, composto por três elementos essenciais:

• **Instruções:** Executadas para fornecer atributos e funções de desempenho desejados.

• **Estruturas de dados:** Permitem a manipulação adequada das informações.

• **Documentação:** Informações descritivas sobre a operação, diagramas e funcionalidades.

É fundamental distinguir **software** de **sistema**. Software é um programa de computador, uma sequência lógica de instruções. Já um sistema é um conjunto de componentes inter-relacionados (software, hardware e recursos humanos) que funcionam de forma unificada para atingir um objetivo comum.

A necessidade de melhorar os softwares é evidente, uma vez que as demandas dos clientes evoluem rapidamente com a tecnologia. A partir de 2010, empresas já necessitavam de sites e aplicativos com banco de dados na nuvem, e, a partir de 2020, a Inteligência Artificial passou a ser considerada para otimizar processos.

Ao contrário do hardware, o software não se desgasta fisicamente, mas pode deteriorar-se com as mudanças e acréscimos de funcionalidades, o que pode aumentar a taxa de defeitos ao longo do tempo. As Leis de Lehman, desenvolvidas na década de 1980, ainda são aplicáveis para verificar a dinâmica da evolução dos softwares, destacando a necessidade de adaptação contínua para evitar a ineficácia.

**2.2. A Prática da Engenharia de Software**

A engenharia de software abrange processos, métodos e ferramentas que permitem a construção de sistemas. Pressman (2016) organiza a engenharia de software em quatro camadas:

• **Qualidade:** Um alicerce para a engenharia de software, garantindo a excelência do produto.

• **Processo:** Fundamento que define o arcabouço para um conjunto de áreas-chave.

• **Métodos:** Fornecem informações técnicas e envolvem tarefas como comunicação, análise de requisitos, modelagem, codificação, testes e manutenção.

• **Ferramentas:** Proporcionam suporte automatizado ou semiautomatizado para o processo e métodos, integrando informações entre si.

O objetivo da engenharia de software é que o desenvolvimento seja eficiente, considerando custo, qualidade e tempo. Além disso, Pressman (2016) enfatiza que componentes reutilizáveis devem ser projetados e integrados em novos sistemas.

**2.3. Princípios da Análise de Sistemas**

A análise de sistemas baseia-se em métodos e técnicas de investigação e especificação para encontrar a melhor solução para problemas computacionais, a partir das funcionalidades levantadas. As fases generalizadas da análise de sistema, segundo Pressman (2005), incluem:

• **Análise:** Estudo de viabilidade, definição de funcionalidades e escopo, alocação de recursos e orçamento.

• **Projeto:** Definição lógica do software, layouts de telas e relatórios, estrutura de banco de dados e diagramas gráficos.

• **Implementação:** Codificação do software.

• **Testes:** Procedimentos para encontrar erros e verificar funcionalidades.

• **Documentação:** Registro de todos os processos e diagramas, servindo como ferramenta de comunicação e parte de contrato.

A não adoção de técnicas de engenharia de software na análise de sistemas pode levar a problemas como softwares com manutenção onerosa, falta de padronização na documentação, falta de controle na sequência das tarefas e previsões imprecisas de prazo e custo.

**2.4. O Papel do Analista de Sistemas**

O analista de sistemas é um profissional fundamental na engenharia de software, responsável por pesquisas, planejamentos, coordenação de equipes e recomendação de alternativas de software. Suas tarefas incluem descobrir o que um sistema deve fazer, entender e avaliar as necessidades e expectativas dos usuários, e organizá-las, especificá-las e documentá-las. O analista de sistemas desenvolve especificações, funcionalidades e transações para atender às solicitações dos usuários, necessitando de conhecimento em diversas áreas de negócio e proatividade para buscar informações. Habilidades como conhecimento tecnológico atualizado, organização, visão gerencial e bom relacionamento interpessoal são desejáveis para este profissional.

**3. O Processo de Software**

Um Processo de Software é um conjunto de atividades e resultados relacionados que levam à produção de um software. Ele não é uma determinação rígida, mas uma abordagem adaptável que permite à equipe escolher os processos que melhor se encaixam na filosofia da empresa.

**3.1. Estrutura de um Processo Genérico de Software**

Pressman (2016) afirma que a estrutura de um processo genérico de software é composta por cinco atividades metodológicas essenciais:

• **Comunicação:** Entendimento dos objetivos do projeto e dos requisitos do software através da interação com os envolvidos.

• **Planejamento:** Criação de um "mapa" do projeto, incluindo tarefas técnicas, riscos, recursos, produtos e cronograma.

• **Modelagem:** Criação de modelos e diagramas para melhor compreensão das necessidades do software, auxiliando na codificação e validação.

• **Construção:** Codificação do software com base nos modelos e realização de testes para validar o código gerado.

• **Entrega:** O software é entregue (parcial ou totalmente) para testes do cliente e feedback, com adaptações e correções realizadas durante um período acordado.

Sommerville (2011) também identifica quatro atividades fundamentais em toda a produção de software: Especificação, Projeto e Implementação, Validação e Evolução.

**3.2. Fluxo de Processo**

O fluxo de processo descreve como as atividades metodológicas de cada processo são organizadas. Os principais fluxos de processo incluem:

• **Fluxo de Processo Linear:** Atividades sequenciais, como no Modelo Cascata, onde uma fase só inicia após a conclusão da anterior.

• **Fluxo de Processo Iterativo/Evolucionário:** As atividades são executadas em ciclos, produzindo versões cada vez mais completas do software incrementalmente. Modelos como Prototipação e Espiral são exemplos.

• **Fluxo de Processo Paralelo:** Duas ou mais atividades podem ser executadas simultaneamente, por exemplo, comunicação e análise.

O modelo evolucionário é particularmente alinhado a projetos onde os requisitos do negócio e do produto são instáveis e podem mudar ao longo do tempo.

**3.3. Avaliação e Melhoria do Processo de Software**

O Processo de Software deve ser continuamente avaliado para garantir que atenda a critérios básicos de qualidade. Abordagens como SCAMPI, CBA IPI, SPICE e ISO 9001:2000 para software são utilizadas para avaliação e aperfeiçoamento. Por exemplo, o SCAMPI, baseado no CMMI, fornece um modelo de avaliação em cinco etapas, definindo regras para objetividade na classificação das avaliações.

**4. Conclusão**

A compreensão dos fundamentos e processos da engenharia de software é essencial para profissionais da área de TI. Desde a conceituação do software e a clareza de seus requisitos até a escolha do modelo de processo mais adequado e a adoção de práticas de avaliação contínua, cada etapa contribui para a entrega de um produto final que atenda às expectativas do cliente com qualidade e dentro das restrições de custo e tempo. O papel do analista de sistemas é central nesse ecossistema, garantindo a comunicação, planejamento e execução eficazes ao longo do ciclo de vida do software.

--------------------------------------------------------------------------------

**Referências**

ABPMP. **Association Of Business Process Management Professionals International.** BPM CBOK®: guia para o gerenciamento de processos de negócio corpo comum de conhecimento v. 3.0. Brasília: ABPMP Brasil, 2013.

ENGHOLM JR., H. **Engenharia de Software na Prática.** São Paulo: Novatec, 2010.

FALBO, R. A. **Engenharia de Requisitos:** Notas de Aula. Universidade Federal do Espírito Santo, Vitória, 2012.

FOWLER, M. **UML Essencial:** um breve guia para a linguagem de modelagem de objetos. Tradução: João Tortello. 3. ed. Porto Alegre, Bookman, 2005.

GONÇALVES, E. J. T.; CORTÉS, M. I. **Análise e Projeto de Sistemas.** 3. ed. Fortaleza, CE: EdUECE, 2015.

IBM. **Rational Unified Process: Best practices for software development teams.** From the developer Works archives. IBM Staff, jul. 2005. Disponível em: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251\_bestpractices\_TP026B.pdf. Acesso em: 7 dez. 2019.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **El Proceso Unificado de Desarrollo de Software.** [S.l.]: Addison-Wesley, 2000.

LOPES, R. **Apontamentos de Análise e Modelação de Sistemas.** 1. ed. Universidade de Aveiro, 2016.

PAULA FILHO, W. P. **Engenharia de software: produtos.** 4. ed. Rio de Janeiro: LTC, 2019.

PFLEEGER, S. L. **Engenharia de software: teoria e prática.** Tradução de Dino Franklin. Revisão técnica Ana Regina Cavalcanti da Rocha. 2. ed. São Paulo: Prentice Hall, 2004.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional.** Tradução de João Eduardo Nóbrega Tortello. Revisão técnica: Reginaldo Arakaki, Julio Arakaki e Renato Manzan de Andrade. 8. ed. Porto Alegre: AMGH, 2016.

PRESSMAN, R. S. **Software engineering: a practitioner’s approach.** 6. ed. Nova York: McGraw-Hill, 2005.

ROTH, R. M.; DENNIS, A.; WIXOM, B. H. **Análise e projeto de sistemas.** 5. ed. Rio de Janeiro: LTC, 2014.

SOMMERVILLE, I. **Engenharia de software.** 9. ed. São Paulo: Pearson Prentice Hall, 2011.

TUCKER, A. B.; NOONAN, R. E. **Programming Languages: Principles and Paradigms.** 2. ed. New York: McGraw-Hill Education, 2010.

VALLE, R.; OLIVEIRA, S. B. de. **Análise e modelagem de processos de negócio: foco na notação BPMN.** São Paulo: Atlas, 2013.

VIEIRA, S. R. C. **Remo: uma técnica de elicitação de requisitos orientada pela modelagem de processos de negócios.** 2012. 129 f. Dissertação (Mestrado em Informática) − Universidade Federal do Amazonas (UFAM), Manaus, 2012.

WERLICH, C.; FEDOZZI, R.; SILVA, S. G. da. **Análise e modelagem de sistemas.** Londrina: Editora e Distribuidora Educacional S.A., 2020.