

Revisão: Desenvolvimento Profissional de Software

1. Definição e Diferença Fundamental

O **desenvolvimento profissional de software** é uma atividade distinta da programação amadora ou pessoal. Enquanto qualquer pessoa pode escrever programas para uso próprio, o software profissional é criado por equipes para atender a propósitos de negócio, ser incluído em outros produtos ou vendido no mercado.

A principal diferença é que o software profissional não se resume apenas ao código. Ele engloba um conjunto de elementos, incluindo:

- **Múltiplos programas** que operam em conjunto.
- **Arquivos de configuração** para ajustar o funcionamento do sistema.
- **Documentação do sistema** (que descreve sua arquitetura e estrutura).
- **Documentação do usuário** (manuais e guias de uso).

A **Engenharia de Software** é a disciplina que fornece as técnicas e os métodos para apoiar essa produção profissional, focando na especificação, no projeto, na validação e na evolução de sistemas de software complexos.

2. Tipos de Produtos de Software

Existem duas categorias principais de produtos de software desenvolvidos profissionalmente:

- **Produtos Genéricos:** São sistemas desenvolvidos para serem vendidos no mercado aberto a qualquer cliente. A empresa desenvolvedora detém o controle sobre as especificações e funcionalidades do produto.
 - **Exemplos:** Processadores de texto, planilhas, sistemas de contabilidade, aplicativos gráficos.
- **Produtos Sob Encomenda (Customizados):** São sistemas encomendados por um cliente específico para atender a uma necessidade particular. Nesse caso, o cliente é quem define e controla as especificações do software.
 - **Exemplos:** Sistemas de controle de tráfego aéreo, software para controle de dispositivos eletrônicos específicos, sistemas para um processo de negócio exclusivo de uma empresa.

Atualmente, a distinção entre os dois tipos está se tornando menos nítida, com a popularização de sistemas híbridos, como os ERPs (ex: SAP), que são produtos genéricos amplamente customizados para cada cliente.

3. Atributos Essenciais de um Bom Software

A qualidade de um software profissional vai além de sua funcionalidade básica. Ela é medida por um conjunto de atributos que garantem seu bom desempenho, longevidade e adequação ao usuário. Os atributos essenciais são:

- **Manutenibilidade:** O software deve ser projetado de forma que possa ser facilmente modificado e atualizado para atender a novas necessidades dos clientes ao longo do tempo.
- **Confiança e Proteção:** Inclui confiabilidade (não causar danos em caso de falha), proteção (segurança contra acesso malicioso) e segurança geral do sistema.
- **Eficiência:** O software não deve desperdiçar recursos do sistema, como memória e capacidade de processamento, garantindo bom tempo de resposta e processamento.
- **Aceitabilidade:** O software deve ser compreensível, fácil de usar e compatível com outros sistemas que o público-alvo utiliza.

4. A Disciplina de Engenharia de Software

Engenharia de Software é uma disciplina de engenharia que abrange **todos os aspectos da produção de software**, desde a especificação inicial até a manutenção pós-entrega. Ela não se limita aos processos técnicos, incluindo também o gerenciamento de projetos e o desenvolvimento de ferramentas e métodos.

A abordagem é sistemática e organizada, buscando entregar resultados de qualidade dentro do **cronograma e orçamento**, o que muitas vezes exige compromissos e soluções práticas, mesmo na ausência de teorias perfeitas.

Atividades Fundamentais do Processo de Software

Todo processo de software envolve quatro atividades essenciais:

1. **Especificação:** Clientes e engenheiros definem o que o software deve fazer e suas restrições operacionais.
2. **Desenvolvimento:** O software é projetado e programado.
3. **Validação:** O software é verificado para garantir que atende às necessidades do cliente.
4. **Evolução:** O software é modificado para se adaptar a novas exigências do cliente e do mercado.

Principais Desafios e Aspectos Gerais

A engenharia de software moderna enfrenta três grandes desafios:

- **Heterogeneidade:** Desenvolver software que funcione em diversas plataformas (desktops, móveis, sistemas antigos).
- **Mudança de Negócio e Social:** Entregar e adaptar software rapidamente para acompanhar a velocidade das mudanças no mercado.
- **Segurança e Confiança:** Garantir que os sistemas sejam robustos contra ataques e que a informação do usuário seja protegida.

Diversidade e Fundamentos

Não existe um método único de engenharia de software que sirva para todos os projetos. A abordagem varia drasticamente dependendo do tipo de aplicação (ex: jogos, sistemas de controle embutidos, aplicações web). Apesar dessa diversidade, existem fundamentos que se aplicam a todos os tipos de software:

- O desenvolvimento deve seguir um **processo gerenciado e compreendido**.
- **Confiança e desempenho** são cruciais.
- Os **requisitos** do software devem ser bem compreendidos e gerenciados.
- Deve-se fazer o melhor uso possível de **recursos existentes**, reutilizando software sempre que apropriado.

5. Estudos de Caso Ilustrativos

Para demonstrar a aplicação prática dos conceitos de engenharia de software, são utilizados três estudos de caso distintos, cada um representando um tipo diferente de sistema com desafios específicos:

- **Sistema de Controle de Bomba de Insulina (Sistema Embutido):**
 - **Descrição:** Um sistema médico crítico que controla um dispositivo de hardware (bomba de insulina) para fornecer doses controladas de insulina a pacientes diabéticos. O software coleta dados de um sensor de glicose e aciona a bomba.
 - **Desafios Principais:** Segurança crítica (falhas podem causar danos graves à saúde), disponibilidade (deve funcionar quando necessário) e confiabilidade (deve fornecer a dose correta).
- **Sistema de Informação de Pacientes (MHC-PMS - Sistema de Informação):**
 - **Descrição:** Um sistema para gerenciar registros de pacientes em tratamento de saúde mental. Armazena dados de pacientes, consultas e tratamentos, além de gerar relatórios gerenciais e clínicos.
 - **Desafios Principais:** Privacidade (confidencialidade dos dados dos pacientes é crucial), segurança (alertar sobre pacientes potencialmente perigosos) e disponibilidade, equilibrando a necessidade de acesso com a proteção dos dados.
- **Estação Meteorológica no Deserto (Sistema de Coleta de Dados):**
 - **Descrição:** Um sistema que coleta dados de vários sensores (temperatura, pressão, vento, etc.) em locais remotos e os transmite para um sistema central.
 - **Desafios Principais:** Confiabilidade em ambientes hostis, gerenciamento de energia (autossuficiência com baterias e recarga solar/eólica) e manutenibilidade remota (atualizações de software e reconfiguração dinâmica).

6. Pontos-Chave

- **Engenharia de software** é uma disciplina que se preocupa com todos os aspectos da

produção de software.

- **Software** não é apenas o programa, mas também toda a sua **documentação**. Seus atributos principais são: manutenibilidade, confiança, proteção, eficiência e aceitabilidade.
- O **processo de software** inclui todas as atividades do seu desenvolvimento: especificação, desenvolvimento, validação e evolução.
- As **ideias fundamentais** da engenharia de software (processos, confiança, requisitos, reuso) são universalmente aplicáveis a todos os tipos de sistemas.
- Existem **vários tipos de sistemas**, e cada um requer técnicas e ferramentas de engenharia de software adequadas. Não há uma solução única para todos os casos.
- Engenheiros de software têm **responsabilidades profissionais e sociais**, que vão além das questões puramente técnicas.
- Sociedades profissionais publicam **códigos de conduta** para definir os padrões de comportamento esperados de seus membros.

Questões sobre o conteúdo

1. O que diferencia fundamentalmente o software profissional da programação pessoal ou amadora?

- a) O uso de linguagens de programação mais recentes e poderosas.
- b) O fato de ser desenvolvido por equipes para atender a propósitos de negócio, em vez de uso próprio.
- c) A ausência total de bugs ou falhas no produto final.
- d) A obrigatoriedade de ser um software de código aberto.

2. Além do código-fonte, o que mais compõe um produto de software profissional?

- a) Apenas o arquivo executável.
- b) O marketing do produto e as estratégias de venda.
- c) Documentação do sistema, documentação do usuário e arquivos de configuração.
- d) O hardware no qual o software será executado.

3. Qual é a principal função da Engenharia de Software como disciplina?

- a) Focar exclusivamente na escrita de código limpo e otimizado.
- b) Fornecer técnicas e métodos para apoiar a produção profissional de software complexo.
- c) Gerenciar equipes de marketing para a venda de produtos de software.
- d) Desenvolver novos sistemas operacionais para computadores.

4. Um aplicativo de edição de fotos desenvolvido por uma empresa para ser vendido em lojas de aplicativos para qualquer usuário é um exemplo de:

- a) Produto Sob Encomenda
- b) Software de Nicho
- c) Produto Genérico
- d) Sistema Legado

5. Quem geralmente define e controla as especificações de um produto de software "Sob Encomenda"?

- a) A equipe de desenvolvimento.

- b) A empresa que está desenvolvendo o software.
 - c) O cliente que encomendou o sistema.
 - d) O mercado e as tendências de tecnologia.
6. Sistemas de ERP (como o SAP), que são produtos genéricos, mas amplamente customizados para cada cliente, são considerados:
- a) Produtos exclusivamente genéricos.
 - b) Produtos exclusivamente customizados.
 - c) Sistemas híbridos, que mesclam características dos dois tipos.
 - d) Softwares de baixo custo e pouca complexidade.
7. O atributo de "Manutenibilidade" em um software refere-se à:
- a) Facilidade com que o software pode ser modificado e atualizado para atender a novas necessidades.
 - b) Capacidade do software de se manter funcionando sem a necessidade de atualizações.
 - c) Complexidade do código, que exige equipes de manutenção especializadas.
 - d) Necessidade de manutenção constante do hardware onde o software roda.
8. O que o atributo "Confiança e Proteção" engloba?
- a) Apenas a garantia de que o software não possui vírus.
 - b) A velocidade de processamento e o tempo de resposta do sistema.
 - c) Confiabilidade (não causar danos em falhas), proteção contra acesso malicioso e segurança geral.
 - d) A popularidade e a boa reputação da marca do software no mercado.
9. Um software que não desperdiça recursos como memória e processador possui alta:
- a) Aceitabilidade
 - b) Eficiência
 - c) Manutenibilidade
 - d) Proteção
10. A "Aceitabilidade" de um software está relacionada a:
- a) Ser o software mais barato disponível no mercado.
 - b) Ter o código-fonte mais complexo e robusto.
 - c) Ser compreensível, fácil de usar e compatível com outros sistemas do usuário.
 - d) Estar disponível em todas as plataformas e sistemas operacionais existentes.
11. Qual das seguintes afirmações descreve melhor a abrangência da Engenharia de Software?
- a) É uma disciplina focada apenas nos aspectos técnicos da programação.
 - b) É uma disciplina focada exclusivamente no gerenciamento de projetos.
 - c) Abrange todos os aspectos da produção de software, da especificação à manutenção.
 - d) Trata-se do estudo teórico da ciência da computação, sem aplicação prática.
12. Na atividade de "Especificação" do processo de software, o que é definido?
- a) A linguagem de programação e as ferramentas que serão utilizadas.
 - b) O cronograma do projeto e o orçamento disponível.
 - c) O que o software deve fazer e suas restrições operacionais.
 - d) A equipe de desenvolvimento e os papéis de cada membro.
13. A atividade de "Validação" tem como principal objetivo:

- a) Vender o software para o maior número de clientes possível.
 - b) Escrever o código-fonte do sistema.
 - c) Verificar se o software atende às necessidades definidas pelo cliente.
 - d) Modificar o software para adicionar novas funcionalidades não planejadas.
14. A fase de "Evolução" em um processo de software lida com:
- a) Apenas a correção de bugs encontrados após o lançamento.
 - b) A criação da primeira versão funcional do software.
 - c) O treinamento dos usuários para utilizarem o sistema.
 - d) A modificação do software para se adaptar a novas exigências do cliente e do mercado.
15. Qual dos grandes desafios da engenharia de software moderna está relacionado ao desenvolvimento de sistemas que funcionem em desktops, celulares e plataformas antigas?
- a) Mudança de Negócio e Social
 - b) Segurança e Confiança
 - c) Heterogeneidade
 - d) Reúso de código

Gabarito

- 1. **b)** O fato de ser desenvolvido por equipes para atender a propósitos de negócio, em vez de uso próprio.
- 2. **c)** Documentação do sistema, documentação do usuário e arquivos de configuração.
- 3. **b)** Fornecer técnicas e métodos para apoiar a produção profissional de software complexo.
- 4. **c)** Produto Genérico
- 5. **c)** O cliente que encomendou o sistema.
- 6. **c)** Sistemas híbridos, que mesclam características dos dois tipos.
- 7. **a)** Facilidade com que o software pode ser modificado e atualizado para atender a novas necessidades.
- 8. **c)** Confiabilidade (não causar danos em falhas), proteção contra acesso malicioso e segurança geral.
- 9. **b)** Eficiência
- 10. **c)** Ser compreensível, fácil de usar e compatível com outros sistemas do usuário.
- 11. **c)** Abrange todos os aspectos da produção de software, da especificação à manutenção.
- 12. **c)** O que o software deve fazer e suas restrições operacionais.
- 13. **c)** Verificar se o software atende às necessidades definidas pelo cliente.
- 14. **d)** A modificação do software para se adaptar a novas exigências do cliente e do mercado.
- 15. **c)** Heterogeneidade