

Atividade 01 – Conceitos Iniciais

1. O que é Software?

Software é uma sequência de comandos e instruções escritos em uma determinada [linguagem de programação](#). No qual estes comandos, instruções, é quem permitem o seu funcionamento.

2. O que é engenharia de software?

A [engenharia de software](#) permite a utilização de teorias, técnicas e ferramentas, assim possibilitando a produção e desenvolvimento de sistemas por meio da análise, coleta e processamento de dados. Atribuindo-se a identificação de potenciais falhas nesses produtos e criam soluções de alta performance.

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

A [engenharia de software](#) se concentra nos aspectos práticos da produção de um sistema de software, enquanto a [ciência da computação](#) estuda os fundamentos teóricos dos aspectos computacionais.

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

A [Engenharia de Software](#) é uma área centralizada na produção de [software](#). Este último termo remete à parte lógica do computador, ou seja, é um conjunto de instruções que são interpretadas por dispositivos para executar suas funções específicas. Por outro lado, a área de [Sistema de Informação](#) está relacionada ao planejamento e implementação de sistemas. O profissional pode ser responsável ainda pela construção de programas informatizados, além de ter como tarefa o planejamento da infraestrutura de TI das empresas.

5. O que é processo de software?

Segundo Sommerville “Um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software” Existem diferentes maneiras de organizarmos as atividades que juntas levam a construção de software. Assim, é possível montar diferentes processos de software.

6. O que é um modelo de processo de software?

Um modelo de processo pode ser visto como uma representação, ou abstração dos objetos e atividades envolvidas no processo de software.

Exemplos de alguns principais modelos de processo de software:

- [Modelo de ciclo de vida.](#)
- [Sequencial ou Cascata](#)
- [Evolucional ou Prototipação](#)
- [Espiral](#)

7. Quais são os custos da engenharia de software?

Custos de software em geral dominam os custos do sistema. Os custos de software em um PC são, em geral, maiores que o custo do hardware. Em Software, custa-se mais para manter do que para desenvolver. Para sistemas de vida útil longa, os custos de manutenção podem ser várias vezes mais que o custo de desenvolvimento. A engenharia de software está preocupada com o desenvolvimento de software com uma boa relação custo-benefício.

8. Quais são os métodos da engenharia de software?

8.1 Metodologia Estruturada

Análise Estruturada, Projeto Estruturado, Programação Estruturada, Análise Essencial, SADT, DFD - Diagrama de Fluxo de Dados, MER - Modelo de Entidades e Relacionamentos

8.2 Metodologia Orientada a Objetos

Orientação a Objetos, Rational Unified Process (RUP), Desenvolvimento ágil de software, Feature Driven Development (FDD), Enterprise Unified Process (EUP), Scrum (Scrum), Crystal (Crystal Clear, Crystal Orange, Crystal Orange Web), Programação extrema (XP)

9. O que é CASE (Computer-Aided Software Engineering)?

As ferramentas do CASE são usadas por gestores de projeto, por analistas e por coordenadores do software para desenvolver o sistema de software. Há um número de ferramentas do CASE disponíveis para simplificar várias fases do ciclo de vida da programação de software tais como ferramentas de análise, ferramentas de projeto, ferramentas de gestão do projeto, ferramentas de gestão do base de dados e ferramentas da documentação.

10. Quais são os atributos de um bom software?

10.1 Funcionalidade

A funcionalidade de um software diz respeito à satisfação de necessidades que deram origem ao projeto. Abrange requisitos implícitos e explícitos e está intimamente ligada à qualidade do código criado.

10.2 Usabilidade

O software precisa ser de fácil utilização, principalmente porque a grande maioria dos usuários não entende de linguagem de programação.

10.3 Eficiência

Um nível de desempenho mínimo e máximo para o software devem ser delimitados no início do projeto. Assim, é possível mapear o comportamento da solução em relação aos recursos que vai utilizar e ao tempo de resposta para o processamento. Tudo deve ser compatível com o nível que foi requerido para o produto.

10.4 Manutenibilidade

A facilidade com que as correções podem ser implementadas também diz muito sobre a qualidade do software. Isso porque permite atualizações rápidas e pouco onerosas, evitadas pelos usuários que temem pela integridade das informações que já foram armazenadas no banco de dados.

10.5 Portabilidade

Essa característica se relaciona com a capacidade de escalabilidade da solução, assim como a necessidade de menor dependência do desenvolvedor que criou o código principal. Para isso, ele precisa ser adaptável a qualquer ambiente sem a necessidade de outras aplicações e estar de acordo com padrões de portabilidade para ser substituído se necessário.

11. Quais são os desafios-chave enfrentados pela engenharia de software?

- Lidar com sistemas legados, lidar com a diversidade crescente e lidar com a crescente demanda e reduzir o tempo de entrega.
- Sistemas legados
- Sistemas antigos, porém, úteis devem ser mantidos e atualizados
- Heterogeneidade
- Sistemas são distribuídos e inclui uma mistura de hardware e software. Deve-se desenvolver técnicas para construir softwares confiáveis e flexíveis.
- Fornecimento
- Existe pressão crescente para uma entrega rápida do software.

