

Atividade 02, Part.1 – Conceitos de Ciclos de Vida

1. O que é software? Comente sobre seus componentes

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. Basicamente componentes de software é um pedaço de software independente, mas que pode ser usado por outros softwares.

2. O que é produto de software?

Um produto de software consiste em uma estratégia de realizar o reuso de forma sistemática para a construção de sistemas com menos esforço desde que estes pertençam a uma mesma família, ou seja, que tenham em comum pertencer a um mesmo domínio de mercado.

3. Diferencie produto genérico de produto sob encomenda

Atualmente, é comum que uma empresa que não seja do ramo da tecnologia tenha uma demanda tecnológica que seja específica o suficiente para que os softwares de prateleira no mercado não atendam. Os softwares sob demanda são soluções projetadas especificamente para uma empresa em particular, que participa do processo de criação, garantindo um alinhamento entre as expectativas e necessidades da empresa e o produto final, já os chamados softwares de prateleira, em que o cliente simplesmente compra a licença do software no mercado, recebendo soluções genéricas e pré-estabelecidas.

4. 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 criando soluções de alta performance.

5. O que são problemas fundamentais e acidentais da Engenharia de Software?

Segundo [Brooks](#), existem dois tipos de dificuldades em desenvolvimento de software: dificuldades essenciais e dificuldades acidentais. As essenciais são da natureza da área e dificilmente serão superadas por qualquer nova tecnologia ou método que se invente.

Segundo Brooks, as dificuldades essenciais são as seguintes:

- **Complexidade:** dentre as construções que o homem se propõe a realizar, software é uma das mais desafiadoras e mais complexas que existe. Na verdade, como dissemos antes, mesmo construções de engenharia tradicional, como um satélite, uma usina nuclear ou um foguete, são cada vez mais dependentes de software.
- **Conformidade:** pela sua natureza software tem que se adaptar ao seu ambiente, que muda a todo momento no mundo moderno. Por exemplo, se as leis para recolhimento de impostos mudam, normalmente espera-se que os sistemas sejam rapidamente adaptados à nova legislação.
- **Facilidade de mudança:** consiste na necessidade de evoluir sempre, incorporando novas funcionalidades. Na verdade, quanto mais bem sucedido for um sistema de software, mais demanda por mudanças ele recebe.
- **Invisibilidade:** devido à sua natureza abstrata, é difícil visualizar o tamanho e consequentemente estimar o esforço de construir um sistema de software.

6. O que são sistemas legados?

Sistemas legados são desenvolvidos para ter uma longa vida útil, sendo organizados em modelo de camadas (sistema sociotécnico). Os sistemas legados por conta do tempo começam a se tornar obsoletos dentro das rotinas em uma organização, por exemplo, um software que foi feito há uma década. Suas funcionalidades serão fortemente impactadas em virtude de muitas mudanças.

7. 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.

8. Quais as atividades fundamentais do processo de software?

Segundo Sommerville, as quatro atividades que devem estar incluídas em qualquer processo de software são: especificação, projetos e implementação, validação e evolução.

9. Na atividade de especificação, qual o papel do estudo de viabilidade?

O objetivo de um estudo de viabilidade é avaliar sob o ponto de vista operacional, técnico, econômico e organizacional se o projeto é viável. O estudo de viabilidade acontece após a especificação de requisitos de negócio, ou seja, é o segundo passo do processo de engenharia de requisitos. Na hora de analisar a viabilidade operacional é essencial utilizar um framework para categorizar problemas (checklist) e auxiliar no levantamento de requisitos, chamado PIECES.

10. Na atividade de projeto e implementação, qual a diferença entre os projetos de arquitetura, interface e componentes?

A **Arquitetura** é o relacionamento entre os grandes componentes estruturais do programa é definido, já a **interface** é o sistema de mecanismos de interação e layout para a interação homem-máquina são estabelecidos. Os **componentes** são o modelo de domínio da informação que são transformados nas estruturas de dados que serão exigidas para implementar o software.

11. Na atividade de validação, o que significa o V&V?

O V&V significa os testes que fazem parte do processo de verificação e validação (V&V). No qual devem ser usados em conjunto com a verificação estática (inspeção).

12. Na atividade de validação, qual a diferença entre os testes de sistema, desenvolvimento e aceitação?

O **Teste de Desenvolvimento** é os componentes que são testados pelas pessoas que os desenvolvem, já o **Teste de Sistema** são os componentes integrados para criar um sistema (ou subsistema) são testados, e por fim, o **Teste de Aceitação** é o sistema que é testado com dados do cliente.

13. Discorra sobre a atividade de evolução de software.

Os sistemas geralmente refletem situações do mundo real e, com isso, há uma necessidade que o software mude acompanhando as mudanças de requisitos impostos pelo ambiente em que está inserido. Se o sistema não sofre essas mudanças, pode ficar obsoleto e cair em desuso. O envelhecimento de um software é um processo inevitável, mas é possível de ser compreendido e suas causas previstas, para que sejam minimizados os impactos dos danos causados por esse envelhecimento. As desvantagens causadas pelo envelhecimento de um software são a perda de desempenho devido a modificações não adequadas na sua estrutura interna, geração de novos erros devido a alterações indevidas

no código e perda de usuários devido à falta de meios para concorrer com versões mais recentes de sistemas semelhantes, como por exemplo, a utilização em sistemas operacionais diferentes.

14. Qual foi o motivador para o surgimento da Engenharia de Software?

Antigamente o trabalho era centrado no programador, que a partir do uso de sua criatividade, resulta em produtos únicos que não podiam ser replicados. Neste cenário, percebe-se a pouca utilização de documentação e de boas práticas de engenharia. Como consequência, um dos problemas enfrentados na época foi o tempo necessário para conclusão de um software e as falhas que ele apresentava. Destarte, para entregar produtos sem falhas, no prazo estabelecido e poder replicar o produto de software. A execução desses aspectos, levaram ao surgimento da Engenharia de Software.

15. Trace um paralelo da Engenharia de Software como Engenharia Civil e como Arte.

Tanto na Engenharia de Software quanto na Engenharia Civil é importante seguir processos que levaram a construção de uma obra de qualidade. Por exemplo, o levantamento de requisitos faz com que a obra, um software ou um prédio, esteja de acordo com as exigências do cliente. Enquanto que a Arte precisa ser concisamente bem elabora e estruturado quanto às formas e traços.