

DCEL / CEUNES / UFES	
Disciplina: Engenharia de Software	Prof. Maria das Graças da Silva Teixeira
Conteúdo: Introdução a Software e Engenharia de Software	

Aluno: Christian Jonas Oliveira.

## 1ª. Lista de Exercícios

1 - O software é fator de diferenciação de muitos produtos e sistemas baseados em computador. Apresente exemplos de dois sistemas / produtos em que o software, não o hardware, é o elemento que faz a diferença.

**R:** Um sistema de gerenciamento de alguns bancos por exemplo, usam um software baseado em COBOL como linguagem. Nesse caso o Software está funcionando corretamente e atende as diversas necessidades desse banco, que facilmente poderia pagar por um Hardware novo. Outro produto é o Macintosh, um software dos produtos da Apple que são o diferencial da empresa, por garantir maior segurança, portabilidade entre os seus produtos e versatilidade que no caso nos leva a questão de que o Software é o diferencial.

2 - Nas décadas de 50 e 60, a programação era aprendida num ambiente semelhante ao de aprendizes. Como os primórdios afetaram as práticas de desenvolvimento de software atuais?

**R:** Naquela época o desenvolvimento de software era constantemente suscetível a erros, uma vez que não havia algo sólido a que se fossem fundamentados os “novos” conceitos. Aprendiam na base da tentativa e do erro. Como consequência disto, foram criadas, as tais boas práticas de programação e desenvolvimento, as quais preencheram as lacunas existentes, possibilitando assim que os mesmos erros não fossem cometidos, uma vez que já temos hoje em dia com o que fundamentar.

### 3 - Mitos do Software

Muitos dos problemas com software podem remeter-se a uma mitologia que surgiu nos primórdios do seu desenvolvimento. Ao contrário dos mitos antigos, que muitas vezes oferecem lições humanas que valem a pena ser consideradas, os mitos do software propagaram desinformação e confusão. Para cada um dos mitos apresentados, comente-os mostrando a realidade dos fatos.

#### - Mitos Administrativos

**Mito:** Meu pessoal tem ferramentas de desenvolvimento de software de última geração, afinal foram comprados os mais novos computadores.

**R:** Existem dois problemas neste **MITO**, um deles é a superestimação dos tais computadores. Como sabemos, não necessariamente o hardware é o diferencial na hora de se produzir um software, é necessário conhecimento técnico aprofundado sobre o que está sendo desenvolvido. O outro ponto é o fato da falácia que somente os computadores, “empurrariam” o desenvolvimento. Sabemos que o processo de desenvolvimento envolve principalmente o ambiente de desenvolvimento, a linguagem utilizada dentre outras ferramentas de **DESENVOLVIMENTO**, e não somente o computador em si.

**Mito:** Se um projeto está atrasado nos prazos, basta adicionar mais programadores.

**R:** Neste **MITO**, o ponto errado se dá pelo fato de que, se um projeto está atrasado, demoraria muito mais tempo para capacitar e dividir as tarefas entre os novos programadores do que simplesmente planejar previamente o que deveria ser feito em cada dia para que o prazo fosse cumprido. Um exemplo válido é que, se uma obra de arte (uma pintura de um quadro por exemplo) está atrasada, não se colocam dois pintores com ideias e expressões diferentes para fazê-la ao mesmo tempo.

#### Mitos do Cliente

**Mito:** Uma declaração geral dos objetivos é suficiente para se começar a escrever programas, podemos preencher os detalhes mais tarde.

**R:** Para este **MITO**, pode-se afirmar que essa tal “declaração geral dos objetivos” **NÃO** é suficiente para começar a escrever o programa. É necessário um conhecimento aprofundado das necessidades que o programa deve suprir antes do início do desenvolvimento, garantindo que mais tarde não haja problemas ou incompatibilidades com o que foi desenvolvido, proporcionando um melhor resultado inicial e dando uma boa primeira impressão para o cliente.

DCEL / CEUNES / UFES	
Disciplina: Engenharia de Software	Prof. Maria das Graças da Silva Teixeira
Conteúdo: Introdução a Software e Engenharia de Software	

**Mito:** Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o software é flexível.

**R:** Conectando com o **MITO** anterior, podemos afirmar que a requisição do projeto deve ser feita previamente com o devido planejamento, pois dependendo da escala do software tais mudanças teriam impactos drásticos para o resto do desenvolvimento. O software ser flexível não garante que não esteja suscetível a erros. Com o devido planejamento prévio, esses erros podem ser evitados antes mesmo do início da codificação do mesmo.

#### *Mitos do Profissional*

**Mito:** Assim que escrevermos o programa e o colocarmos em funcionamento o trabalho estará completo.

**R:** A afirmativa é um **MITO** pelo fato de ser necessária uma avaliação completa do que foi feito e passar por testes rigorosos, afim de procurar por erros, bugs, etc. Após isso o produto final é passado ao cliente afim de saber se atende a todos os critérios exigidos.

**Mito:** A única coisa a ser entregue em um projeto bem sucedido é o programa funcionando.

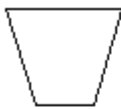
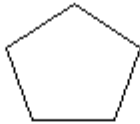



**R:** Não somente o programa, como também a documentação gerada de todo o processo para que em futuras atualizações esteja bem descrito o que cada parte do programa faz, facilitando na identificação do que deve ser mudado.

LISTE MAIS DOIS MITOS E COMENTE-OS.

**Mito:** A documentação gerada no processo de desenvolvimento é inútil.

**R:** Este é um **MITO** comum, principalmente para quem inicia na ramo de desenvolvimento de software, uma vez que não se tem conhecimento da proporção que um software pode tomar. A documentação é fundamental, para que todos os envolvidos no desenvolvimento saibam exatamente o que está ocorrendo em cada fase do projeto e para que futuras atualizações possam ser feitas da melhor forma possível, garantindo assim uma longevidade na qualidade do programa.

4 - Comente a figura abaixo, relacionado ao desenvolvimento de software:

 <p>○ que é necessário fazer</p>	 <p>○ que os usuários querem</p>	 <p>○ que os usuários pedem</p>
 <p>○ que os analistas entendem</p>	 <p>○ que acaba sendo feito</p>	

**R:** A figura ilustra o que uma deficiência na comunicação e na definição do que deve ser feito pode gerar. Ilustra também a dificuldade que é entender o que deve ser feito em cada etapa do processo sem alguém que tome a frente e tenha uma macro visão de todas as fases desse processo. É nesses casos que um bom Engenheiro de Software deve agir, pois assim haverá um alinhamento daquilo que foi pedido e do produto final a ser entregue.

DCEL / CEUNES / UFES	
Disciplina: Engenharia de Software	Prof. Maria das Graças da Silva Teixeira
Conteúdo: Introdução a Software e Engenharia de Software	

5 – Procurou-se formular ao longo do tópico sobre Software uma melhor compreensão deste como um produto, quais as características que ele tem e os principais problemas que está enfrentando / enfrentou. Qual é a sua compreensão a partir de agora sobre o que é um software? Quais as principais características dele como produto? O que diferencia o produto software dos produtos habitualmente produzidos?

**R:** Acredito que podemos ter a definição de software como algo que vem pra desempenhar, organizar ou controlar alguma função, sendo que o mesmo une os componentes lógicos de um computador ou sistema, de maneira que o mesmo é rodado em um computador, celular ou dispositivo que o permita executar funções (calculadora, micro-ondas, TV, etc). Tem como característica a funcionalidade, a confiabilidade, a usabilidade, a eficiência, a portabilidade, a manutenibilidade, entre outras. Sua principal diferença do resto dos produtos é a degradabilidade, uma vez que uma geladeira, um carro, uma moto, um alimento, se degradam com o tempo e o software não.

6 - Quais os problemas que habitualmente são enfrentados por aqueles que se utilizam do software, bem como aqueles que desenvolvem software.

**R:** Problemas como a demora pra elaboração até um produto final e também na manutenção do mesmo, os altos custos envolvidos, os erros e bugs após a conclusão e durante a utilização do usuário, a dificuldade em medir progressos durante o desenvolvimento do software, são comuns na área de desenvolvimento de software, alguns podem ser evitados ou contornados, já outros não há como resolver, pois dependem do produto final para serem encontrados.

7 - Defina com suas palavras Engenharia de Software e quais os principais objetivos a serem alcançados por ela.

**R:** Engenharia de software é entendida por mim como a área da computação responsável pela administração de todos os recursos físicos, humanos e intelectuais durante o desenvolvimento de um software. E não somente isso, é ela também que é responsável pelo planejamento prévio e pela elaboração dos métodos que serão utilizados em todo o processo.

8 - O que são métodos, ferramentas e procedimentos? Como esses conceitos se relacionam?

**R:** Durante o processo de desenvolvimento é necessário a definição desses três conceitos. No meio da Engenharia de Software podemos entendê-los como:

Os métodos – Referem-se as técnicas utilizadas na elaboração do mesmo, isto é, o modo em como será feita cada parte do software.

As ferramentas - são, como o próprio nome já diz, o conjunto de programas, ambientes de desenvolvimento, e tudo que é utilizado para a elaboração prática do software;

Os procedimentos – define a maneira com os quais os métodos serão utilizados.

Esses três conceitos se relacionam de maneira que são dependentes uns dos outros, e há uma ordem cronológica na qual cada um deve ser executado.

9 - Dê exemplo de uma ferramenta CASE (de preferência alguma que você já tenha utilizado). Explique, rapidamente, quais são as suas funcionalidades.

**R:** O MySQL é uma ferramenta CASE. Usei para fazer um trabalho final na disciplina de Programação Orientada a Objetos. Nela é possível guardar informações em banco de dados fornecido pela Oracle.

DCEL / CEUNES / UFES	
Disciplina: Engenharia de Software	Prof. Maria das Graças da Silva Teixeira
Conteúdo: Introdução a Software e Engenharia de Software	

10 – Na pergunta 6 solicitou-se a identificação dos principais problemas do software. Agora, questiona-se: quais são as causas principais desses problemas, e como a Engenharia de Software pode auxiliar.

**R:** Muitos dos problemas se dão pelo mal gerenciamento pré, durante e pós codificação do software, podendo ser evitados pela devida orientação de um profissional da ES.

11 – Comente sobre as 4 fases genéricas da Engenharia.

**R:** Definição: é onde se identifica o problema e busca-se uma maneira de resolvê-lo;

Desenvolvimento: é a fase onde ocorrem as atividades de codificação, design e testes;

Operação: é aqui onde deve-se dar o suporte aos usuários, corrigindo bugs, erros, etc;

Retirada: quando o software deixa de ser utilizado, onde será necessária uma migração, uma mudança total do software.

12 – À medida que o software torna-se mais difundido, os riscos para o público (por causa de programas errados) tornam-se uma preocupação crescente e significativa. Desenvolva um cenário catastrófico realístico em que a falha de um programa de computador pode provocar grande prejuízo (econômico ou humano).

**R:** Como já ocorrido, um momento catastrófico foi a explosão do Mariner I, que gerou um prejuízo de US\$ 800 milhões, causado por um erro de programação, onde um hífen causou tal erro e explosão, fazendo com que a nave ignorasse todos os dados enviados pelo computador até o radar.