**Atividade Prática 6 - Mitos de Desenvolvimento de SW**

**Nome:** Lucas Oliveira de Farias

**RA:** a2371600

**1 – O que foi a chamada “Crise de Software”?**

**R:** Foi um termo, que surgiu em 1970, época em que a engenharia de software quase não existia. Nesse período o desenvolvimento de software enfrentava algumas dificuldades devido ao: Rápido crescimento da demanda por software, aumento da complexidade dos problemas a serem solucionados e a inexistência de técnicas estabelecidas que funcionassem perfeitamente.

**2 - Como a Engenharia de Software contribuiu com a superação da “Crise do Software”?**

**R:** A Engenharia de Software, surgiu com intuito de contornar a crise do software, com a preocupação de sistematizar o processo de desenvolvimento e manutenção do software.

Sendo assim, a engenharia de software incentiva o uso de modelos abstratos e precisos, permitindo especificar, projetar, implementar e manter um sistema, sempre garantindo a qualidade do sistema; oferece planejamento e gerenciamento para os processos do desenvolvimento.

A engenharia também segue alguns princípios, como: Formalidade, Abstração, Decomposição, Generalização e Flexibilização.

Alguns elementos essenciais são aplicados pela engenharia, são eles: Foco na qualidade, Ferramentas, métodos e processo.

**3 - O que são mitos de software?**

Mitos de software são falsas verdades, encontradas no mundo da indústria de software. Tanto os mais jovens, como os mais velhos, dão créditos a essa mentira.

**4 – Pesquise, cite e comente sobre 5 mitos de software**

Alguns dos mitos são:

* **“Um bom manual, cheios de regras e padrões, fornecerá a equipe tudo que é necessário saber”.** Na verdade, não é assim que funciona, pois os projetos são diferentes um dos outros, as regras de negócios são diferentes, as pessoas envolvidas nos projetos são diferentes e pensam diferentes umas das outras. Então não há como criar um manual, pois não há como aplicar um manual para todas as situações. Por esse motivo é imprescindível um planejamento, um levantamento de requisitos, para que o projeto seja iniciado.
* **“Caso haja atraso no cronograma ou no projeto, pode ser que seja alocado mais programadores”.** É um mito bastante “utilizando” ainda hoje. Para desenvolver um software existes várias etapas a serem compridas, e isso requer tempo para se executado. Antes de começar um projeto tem de ser alocado a quantidade de pessoas capacitadas necessárias para executá-lo, e determinar um prazo em que realmente seja o suficiente para concluir o projeto. Não adianta alocar um programador que não está familiarizado com a empresa e com os projetos, pois ao fazer isso, atrapalhará outros programadores que já estão no projeto, visto que eles terão que auxiliar o programador que fora recentemente alocado.
* **“Terceirização de trabalho é uma garantia de tranquilidade e nenhum trabalho”.** Terceirizar um trabalho não é uma boa opção. Pois se uma empresa contratou uma outra empresa para criar e desenvolver um projeto, a responsabilidade é totalmente da empresa contratada, qualquer erro, ou qualquer problema que aconteça ou que venha á acontecer no futuro, é de total responsabilidade da empresa contratada. Para realizar manutenções e correções de erros futuros será mais custoso, mais trabalhosos, talvez o código não esteja bem estruturado, ou não esteja documentado e comentado, a equipe não estará familiarizada com o código. Ao terceirizar um serviço não há garantia de qualidade.
* **“Para a construção de um software, não há necessidade de levar em contas os detalhes”.** Na construção de um software, os detalhes são imprescindíveis, independente da complexidade do sistema que será desenvolvido. Ao analisar todos os detalhes, o desenvolvedor está evitando um retrabalho no futuro, pois no momento em que se começa a desenvolver um sistema sem verificar os mínimos detalhes, sem realizar uma análise de requisitos, quase sempre é necessário refazer o código. Entretanto, ao realizar um levantamento detalhado de todas as funcionalidades, não impedirá erros futuros, mas diminuirá a quantidade dos mesmos.
* **“O único produto que será entregue ao cliente é o código executável”.** O programa em tese, geralmente é apenas o “executável”, mas não é bem assim. Quando um programador entrega um software, ele entrega um conjunto de funcionalidades, que muitas vezes pode impactar na vida diária das pessoas, seja em casa, na escola ou no trabalho. E por isso deve ser desenvolvido com qualidade e perfeição.