**Centro Universitário - UniEvangélica**

**Anápolis 18 de fevereiro de 2018.**

**Acadêmico: Éber Lucas de França Oliveira**

**Professora: Renata Dutra Braga**

**Disciplina: Estimativa de Esforço em Projetos de Software**

**Período: 7° período**

**LISTA 1**

**ATIVIDADES COMPLEMENTARES**

**1) Elabore uma resenha, justificando a importância em estimar projetos de software. Cite, pelo menos, quatro fatores que dificultam a estimativa de projetos de software.**

**2) Pesquise e liste, pelo menos, cinco técnicas de estimativas. Descreva, de forma simplificada, sobre cada uma delas.**

**3) Apresente exemplo de um produto (ou serviço) que você pretende trabalhar ao longo da disciplina. Exemplos: App Taxi, Controle de estoque, Painel de Informações Rodoviária/Universidade, Cotação, outros.**

**Métricas e Estimativas de Software - O início de um rally de regularidade**

No artigo “Métricas e Estimativas de Software - O início de um rally de regularidade”, temos uma explanação sobre a área estimativa de Software e também algumas métricas de avaliação de software. Ele nos introduz muito bem a centro do debate nos mostrando quão importante é estar atento às métricas dentro de um projeto de software utilizando como metáfora uma competição de rally em que 50% dos fatores dependem do navegador e esse se compara aos prazos e custos.

O artigo nos explica quão grande é a importância das métricas e estimativas de software que vem se tornando um dos principais tópicos na Engenharia da Informação com a crescente exigência de seus consumidores pela qualidade, rapidez comodidade e baixo custo de implantação e manutenção, é impossível não enxergar tais técnicas como alavanca para um produto de melhor qualidade, com custos adequados.

E também nos mostra que o ato de medir e estimar é a parte mais importante de um projeto de sistema alguns fatos como: a falta de maturidade, o desinteresse das empresas de desenvolvimento de sistemas e a baixa popularidade deste assunto entre os profissionais da área de informática são algumas das principais causas para o insucesso e o alto custo dos sistemas de informação. E que a métrica de software tem como princípios especificar as funções de coleta de dados de avaliação e desempenho, atribuir responsabilidades a toda a equipe envolvida no projeto, reunir dados de desempenho pertencentes à complementação do software, analisar os históricos dos projetos anteriores para determinar o efeito desses fatores e utilizar esses efeitos para pesar as previsões futuras.

O artigo nos apresenta algumas métricas que são utilizadas para a avaliação de prazos e custos com são eles: As Métricas Orientadas ao Tamanho, Métricas Orientadas à Função, Métricas Voltadas para Orientação a Objeto, Estimativa de Tempo, Estimativa de Custo. O Fator Humano, com base no artigo descreverei brevemente todos os tipos de métricas que foram abordadas.

**As Métricas Orientadas ao Tamanho**

Métricas de software orientadas ao tamanho são medidas diretas do software e do processo por meio do qual ele é desenvolvido. Se uma organização de software mantiver registros simples, uma tabela de dados orientada ao tamanho poderá ser criada.

A partir dos dados brutos contidos na tabela, um conjunto de métricas de qualidade e de produtividade orientadas ao tamanho pode ser desenvolvido para cada projeto. Médias podem ser computadas levando-se em consideração todos os projetos.

As métricas orientadas ao tamanho provocam controvérsias e não são universalmente aceitas como a melhor maneira de se medir o processo de desenvolvimento de software. A maior parte da controvérsia gira em torno do uso das linhas de código (LOC) como uma medida-chave. Os proponentes da afeição de linhas de código afirmam que as mesmas são o "artefato" de todos os projetos de desenvolvimento de software que podem ser facilmente contados, que muitos modelos existentes usam LOC ou KLOC (milhares de linhas de código) como entrada-chave e que já existe um grande volume de literatura e de dados baseados nas linhas de código.

**Métricas Orientadas à Função**

Consiste em um método para medição de software do ponto de vista do usuário, que determina de forma consistente o tamanho e complexidade de um software, sob a perspectiva do usuário. Ele dimensiona um software, quantificando a funcionalidade proporcionada ao usuário a partir do seu desenho lógico.

Um dos princípios da análise de pontos-por-função focaliza-se na perspectiva de como os usuários "enxergam" os resultados que um sistema produz. A análise considera as várias formas com que os usuários interagem com o sistema, com os seguintes objetivos:

1. Fornecer medidas consistentes;

2. Medir funcionalidades que o usuário solicita ou recebe;

3. Independência da tecnologia;

4. Método simples.

As métricas orientadas à função apresentam vários benefícios, dentre eles podemos citar o seguintes:

1. Uma ferramenta para dimensionar aplicações;

2. Um veículo para quantificar custo, esforço e tempo;

3. Um veículo para calcular índices de produtividade e qualidade;

4. Um fator de normalização para comparar software.

Tal métrica parece ser útil e funcional para o desenvolvimento tradicional, mas apresenta algumas falhas com o modelo de desenvolvimento em orientação a objeto (OO), pois alguns atributos do design em OO invalidam o cálculo de alguns pontos-por-função.

A métrica de pontos por função foi originalmente projetada para sistemas de informação comerciais. Para acomodar estas aplicações, a dimensão dos dados foi enfatizada para a exclusão de dimensões funcionais e de controle.

**Métricas Voltadas para Orientação a Objeto**

Muitas métricas já foram desenvolvidas para gerações passadas de tecnologia e, em muitos casos, são usadas até para desenvolvimento OO, porém não são muito coerentes, pois a diferença entre sistemas tradicionais e sistemas OO são muito grandes.

Existem várias propostas para métricas OO que levam em consideração as características básicas e interações do sistema como: número de classes, número de cases, número de métodos, médias de métodos, médias de métodos por classe, linhas de código por método, profundidade máxima da hierarquia de classes, a relação existente entre métodos públicos e privados, entre outros.

**Estimativa de Tempo**

A estimativa do esforço é a técnica mais comum para se levantar os custos de qualquer projeto de desenvolvimento de engenharia. Um número de pessoas-dia, pessoas-mês ou pessoas-ano é aplicado à solução de cada tarefa do projeto. Um custo em dólares é associado a cada unidade de esforço e um custo estimado será derivado. Como a técnica LOC (linhas de código) ou FP (pontos-por-função), a estimativa de esforço inicia-se com um delineamento das funções do software obtidas a partir do escopo do projeto.

O planejador estima o esforço (por exemplo, pessoas-mês) que seria exigido para se concluir cada tarefa de engenharia de software para cada função de software. Taxas de mão-de-obra (isto é, custo/esforço unitário) são aplicados em cada uma das tarefas de engenharia de software. Muito provavelmente, a taxa de mão-de-obra irá variar para cada tarefa

**Estimativa de Custo**

O objetivo desta análise é calcular de maneira antecipada todo e qualquer custo que esteja associado ao sistema, tais como: construção, instalação, operação e manutenção.

O custo da construção é um tópico importante, visto que é graças a ele que sabemos o total de todas as pessoas envolvidas no desenvolvimento do sistema, tais como: burocratas, diretores, membros da comunidade usuária, consultores e programadores, membros da auditoria, do controle de qualidade ou da equipe de operações.

**O Fator Humano**

Quando os objetivos para o desenvolvimento de sistemas não são claros, as pessoas passam a deduzir e criar o produto dentro de suas próprias visões, levando a sistemas inadequados para a função do negócio a ser atendida e, consequentemente, a métricas falhas, gerando uma expectativa divergente entre o cliente e os técnicos ressonáveis, isto é, uma estimativa irreal.

As pessoas são sensíveis aos estímulos externos e por meio de tais estímulos são influenciadas suas atitudes e pensamentos. Um analista, ou um grupo de analistas, disposto a estimar o tempo e custo de um projeto não poderia deixar de dar a devida relevância a este fato. Um grupo de pessoas motivado, trabalhando em um ambiente agradável, sem sofrer qualquer tipo de pressão por parte da empresa ou organização produziria muito mais do que um grupo de pessoas sujeitas a condições adversas a estas.

**Conclusão**

É previsto que em um projeto de software se torna necessário medir o tempo que se gastará e o custo de todo o empreendimento. E com esse artigo podemos observar que existem várias métricas para facilitar a definição de todos esses fatores.

Esse artigo nos introduz bem ao o que é e que estimar não é adivinhar que podemos utilizar diversas formas para estimar corretamente para que recursos não sejam usados sem necessidade ou demasiadamente.