



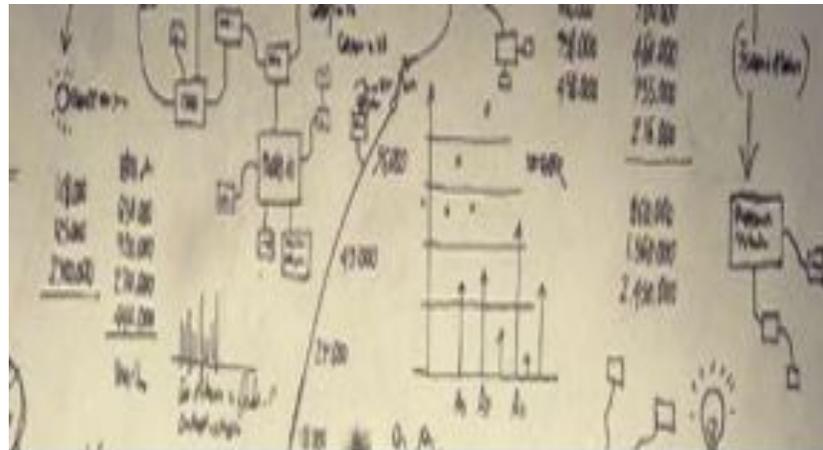
Eixo Tecnológico Informação e Comunicação

Estimativas de Projeto
Rogério Antonio Schmitt



Estimativas de Projeto

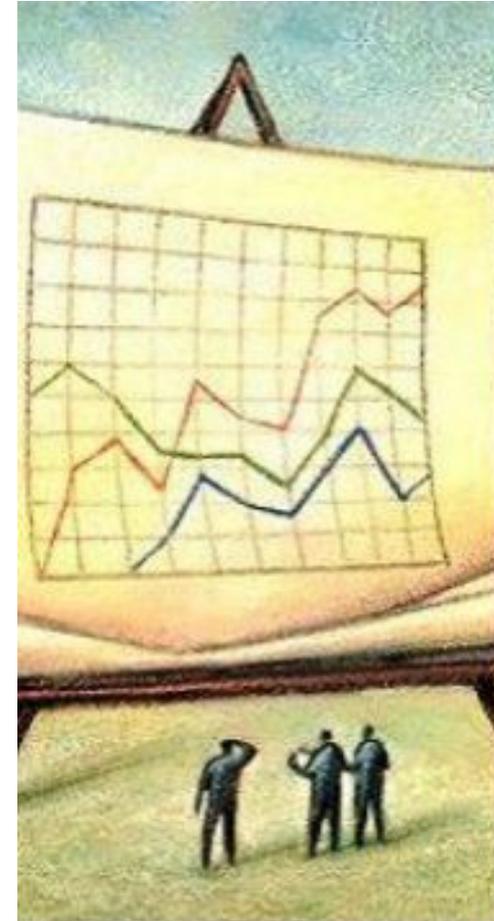
Uma das questões fundamentais em um projeto de software é saber, antes de executá-lo, quanto esforço, em horas de trabalho, será necessário para levá-lo a termo. Essa área, chamada de "estimativa de esforço", conta com algumas técnicas que normalmente têm apresentado bons resultados ao longo do tempo.



**INSTITUTO
FEDERAL**
Santa Catarina

- ▶ O planejamento exige que se assuma um compromisso inicial, mesmo que mais tarde ele venha a se mostrar errado. Sempre que forem feitas estimativas, (e estimativas não são previsões) deve-se olhar o futuro e aceitar um certo grau de incerteza.
- ▶ As estimativas de recursos, custos e cronograma para um trabalho de engenharia de software exigem experiência, acesso a boas informações históricas (métricas) e a coragem de se comprometer com as estimativas

- ▶ Embora a estimativa seja muito mais arte do que ciência, ela não precisa ser conduzida de maneira aleatória. Existem técnicas úteis para estimar tempo e esforço. A maioria das técnicas de estimação de esforço utiliza pelo menos um parâmetro como base, por isso elas são chamadas de **técnicas paramétricas**.



- ▶ Algumas das técnicas paramétricas são baseadas na previsão do número de linhas que o programa deverá ter. Outras, baseiam-se em requisitos, como funções, casos de uso ou histórias de usuário.
- ▶ As técnicas de pontos de função, casos de uso e histórias baseiam-se em um conjunto de requisitos, aos quais é atribuído um peso que determinará o esforço necessário para seu desenvolvimento, a partir de certas transformações matemáticas e tabelas de parâmetros.
- ▶ Um exemplo de técnica não paramétrica é estimar que qualquer projeto de desenvolvimento sobre o qual não se sabe quase nada vai levar seis meses para ser executado.

- ▶ À medida que mais informações sobre o projeto vão sendo disponibilizadas, esse tempo é ajustado para cima ou para baixo, baseando-se na opinião de especialistas. Apesar de essa técnica ser bastante usada, poucos engenheiros de software conseguem obter boas previsões com ela.



- 
- 
- ▶ Peter Drucker, grande especialista em administração, diz que: **“O que pode ser medido, pode ser melhorado”**. Isso se aplica também às estimativas, uma vez que é muito comum serem desenvolvidos recursos ou funcionalidades que são muito semelhantes a algum outro que já foi feito antes. Daí a importância de medir e registrar tudo o que é feito. Ter o registro dessas medidas ajuda a acompanhar a nossa evolução profissional.
- 

- 
- 
- ▶ A experiência na área de desenvolvimento ao longo dos anos ajuda a praticar e a evitar que se "queime o filme" quando for preciso fornecer estimativas, mesmo que se cometa alguns erros de avaliação. Alguns pontos importantes que podem ajudar nessa tarefa de estimativas:

- 1)** Documentar suas tarefas para criar a sua base de conhecimento: Ela vai ser muito útil pra evitar aquele "chute" na hora de fazer uma estimativa. Uma boa estratégia é anotar numa planilha as suas atividades realizadas, guardando o tempo real e o prazo que foi estimado, detalhes de problemas que ocorreram , quais eram os requisitos.





2) Ferramentas gratuitas de controle de projeto, ou controle do tempo podem te ajudar a separar o tempo que é usado para codificar, e o tempo de pesquisa ou uso de ferramentas. Você vai acabar descobrindo quanto tempo do seu dia é realmente usado para programar.

3) Organizar como você faz os commits do código pode ajudar nas próximas estimativas. Usando mensagens claras, curtas e com significado, podem te ajudar a pesquisar o teu processo de liberação de código.



4) Procure usar a "Estimativa de 3 pontos". Ela procura calcular a duração média de uma tarefa utilizando uma estimativa otimista (O), mais provável (MP) e pessimista (P) aplicando o cálculo com pesos diferenciados. É uma técnica bem simples de aplicar.

- ▶ Não existe a melhor técnica, mas a técnica mais adequada a você e ao seu time. Então, se você for gerenciar um projeto, não encare o projeto somente por pressão, ele não pode ser iniciado como uma folha em branco. Se você for desenvolvedor: cobre muito dos requisitos, são eles que vão dar um norte ao seu trabalho.

Referências

Wazlawick, Raul. **Engenharia do Software conceitos e práticas**. Editora Elsevier: 1ª ed, 2013.

