

ESTIMATIVAS DE PROJETO*

Rogério Antônio Schmitt

Objetivos

Ao final deste texto, você deve ser capaz de:

- reconhecer a importância das estimativas e seus impactos em projetos de software;
- identificar técnicas paramétricas e não paramétricas para cálculo de estimativas.

Iniciando o estudo

Um ponto fundamental em projetos de software é procurar determinar, antes da sua execução, quanto esforço, em questão de tempo, será necessário para sua conclusão. Essa área, chamada de "estimativa de esforço", conta com algumas técnicas que normalmente têm apresentado bons resultados ao longo do tempo.

1 Estimativas de projeto

Ao iniciar um projeto, uma boa prática é procurar estabelecer um objetivo inicial, mesmo que ao longo do tempo ele precise ser alterado. Sempre que forem feitas estimativas, (e estimativas não são previsões) olha-se para o futuro, aceitando um certo grau de incerteza.

As estimativas de recursos, custos e cronograma para um trabalho de engenharia de software exigem um certo grau de experiência, além de ser necessário acessar informações históricas, métricas, aplicar técnicas sistemáticas, e também o comprometimento com as estimativas iniciais.

* Texto originalmente publicado na Revista PQANP do IFSC, v.1, n. 1, p.63-66.

Embora não sendo uma ciência exata, a estimativa também não é uma arte aleatória, intuição e experiência podem fazer diferença, e existem técnicas que podem ser utilizadas para facilitar esse processo. 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. Vale destacar que, 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 ainda não se sabe quase nada, vai levar seis meses para ser executado. Com o avanço do projeto, mais informações vão sendo disponibilizadas, e esse tempo é ajustado para cima ou para baixo, baseando-se na opinião de especialistas. Apesar dessa técnica ser bastante utilizada, poucos engenheiros de software conseguem obter boas previsões com ela.

A célebre frase “O que pode ser medido pode ser melhorado” atribuída a Peter Drucker, considerado o pai da administração moderna, pode ser projetada 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 se medir e registrar tudo o que é realizado.

Ter o registro dessas medidas ajuda a acompanhar a nossa evolução no processo de estimar software.

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

GLOSSÁRIO:

Commit: Em ciência da computação e gerenciamento de dados, um commit é a realização de um conjunto de mudanças e proporcionando durabilidade às transações ACID. Fonte: provisórias permanentes, marcando o fim de uma transação e proporcionando durabilidade às transações ACID.

Fonte:

<https://blog.geekhunter.com.br/o-que-e-commit-e-como-usar-commits-semanticos/>

Acesso em: Julho, 2021.

- 1) Documentar suas tarefas para criar uma base de conhecimento sobre projetos já realizados: Ela vai ser muito útil para evitar que se "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 de uma tarefa utilizando uma média com a estimativa otimista (O), mais provável (MP) e pessimista (P) aplicando o cálculo com pesos diferenciados. É uma técnica bem simples de aplicar.

Dentre todas as técnicas, procure identificar a mais adequada a você e ao seu time. Então, se você for gerenciar um projeto, procure não iniciá-lo sem um mínimo de planejamento, como se fosse iniciar do zero. Se você for desenvolvedor: cobre muito dos requisitos, são eles que vão dar um norte ao seu trabalho.

Concluindo o estudo

Não seria muito razoável iniciar um projeto de software sem antes saber o quanto será gasto com ele, uma vez que projetos dessa natureza podem envolver um valor considerável.

Um bom planejamento deve conseguir estimar antecipadamente pelo menos: quanto tempo, quanto esforço e quantas pessoas serão necessárias. Além de, inclusive, o que será necessário em termos de hardware e software, como também os potenciais riscos do projeto.

Como podemos observar, o acesso a dados históricos confiáveis e a aplicação de métodos sistemáticos ajudam a diminuir o grau de incerteza em relação ao

desenvolvimento e a conclusão de projetos de software.

Referência utilizada para a elaboração deste material

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