



Fundamentos Engenharia de Software

UNIDADE 02

Estimativa de Software

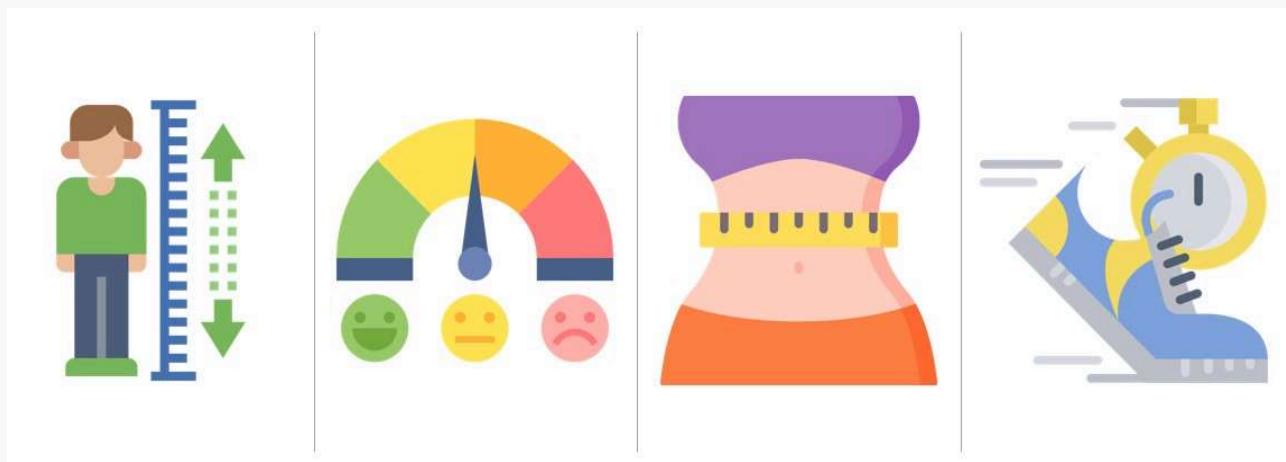
Esta Unidade apresenta as motivações e os conceitos relacionados à estimativa de software e tem início com uma visão geral das atividades de medição e os seus distintos objetivos. Na sequência, será aprofundado o entendimento da atividade de estimativa de esforço, que é muito importante no início do projeto, pois apoia o planejamento, ajudando a identificar os custos e o tempo de desenvolvimento do software. Existem várias técnicas que suportam a atividade de estimativa, desde as mais informais até as bastante estruturadas e parametrizadas. Nesta Unidade, conheceremos algumas delas e nos aprofundaremos em uma técnica muito utilizada nos métodos ágeis, que utiliza como parâmetro principal pontos de histórias do usuário.

| Introdução à medição de *software*

No dia a dia, é comum realizarmos medições para variadas situações. Muitas vezes, nem nos damos conta de como a medição é importante para a qualidade das nossas vidas ou dos produtos e serviços que consumimos. Veja alguns exemplos do porquê medimos (Figura 1):

- Acompanhar a evolução do desenvolvimento de uma criança.
- Monitorar parâmetros de saúde.
- Desempenho em atividades físicas.
- Qualidade do atendimento de um serviço etc.

Figura 1: Motivação para a medição



Motivos para executar atividades de medição em nosso dia a dia. Fonte: Autor (2021).

Se conseguimos entender a importância da medição no nosso dia a dia, também devemos considerar que no desenvolvimento de *software*, a medição é essencial. Mas vamos entender melhor quais são os principais motivos para realizar atividade de medição durante um projeto de desenvolvimento de *software*.

Por que medir o *software*?

A medição no escopo de um projeto de desenvolvimento de *software* é essencial e pode ocorrer em momentos distintos e com objetivos distintos. Alguns motivos para executar atividades de medição são:

+ Entender e aperfeiçoar o processo de desenvolvimento de software

Medir as atividades desenvolvidas ajuda a identificar gargalos e propor melhorias.

+ Melhorar a gerência de projetos, assim como o relacionamento com os clientes

Medir a satisfação do cliente permite ajustes no projeto em andamento.

+ Reduzir frustrações e pressões de cronograma

Ter uma expectativa de execução próxima da realidade estabelece uma relação de confiança com o cliente.

+ Indicar a qualidade de um produto de software

Medir a quantidade de defeitos permite reagir rapidamente a mudanças, em busca de melhorias.

+ Avaliar a produtividade do processo

Medir a produtividade da equipe permite ações assertivas do que

manter ou melhorar.

+ Realizar estimativas

Medir o tamanho do *software* antes do seu desenvolvimento permite projetar cronograma e custos mais próximos da realidade.

As atividades de medição de *software* estão bastante alinhadas com a área de **qualidade de software**. São atividades amplas e, conforme visto, realizadas por diversos motivos. No entanto, nesta disciplina, focaremos especificamente na atividade **estimativas de esforço**, cujo objetivo principal é identificar o esforço de tempo e de custo necessário para se desenvolver um software, antes da sua construção.

Essa atividade acaba sendo mais alinhada com área de **planejamento e gerência de projetos**. Se você possui interesse em aprofundar os conhecimentos nas demais atividades de medição, deve buscar por materiais complementares.

| Estimativas de esforço

Não se pode planejar um projeto sem saber quanto tempo ele vai durar e quanto vai custar. Uma das questões fundamentais em um projeto de *software* é saber, antes de executá-lo, quanto esforço, em horas, dias ou meses de trabalho, será necessário para concluir-lo.

A estimativa é uma atividade desafiadora, pois é difícil saber se é possível desenvolver o produto desejado pelo cliente antes de conhecer os detalhes do projeto. Isso é uma realidade não somente na área de desenvolvimento de *software*. Veja um exemplo na Figura 2, apresentada a seguir, de um cliente que solicita um orçamento para a construção de uma nova casa. No início do projeto pode ocorrer de o cliente não

conseguir verbalizar claramente os requisitos necessários para a sua casa, e o projetista precisar muitas vezes imaginar os desejos do cliente. Esta falta de detalhes, ou até falha de comunicação, pode levar a entendimentos diferentes.

Figura 2: O que o cliente queria, o que o projetista entendeu



Dificuldade em entender os desejos do cliente antes da execução do projeto. Fonte: Autor (2021).

Na área de desenvolvimento de *software*, este detalhamento inicial é ainda mais desafiador pelo fato de o *software* não ser um ativo físico.

Portanto, uma das coisas mais difíceis no desenvolvimento do *software* é estimar o tempo para o desenvolvimento. O *software* é um ativo intangível e de difícil visualização do produto final no início do projeto. A **exatidão** nas estimativas pode ser algo **impossível** de atingir uma vez que pode ser afetado por muitas variáveis, tais como fatores humanos, técnicos, ambientais, políticos etc.

Determinar o esforço e o custo é uma das primeiras e principais atividades relacionadas às estimativas a serem efetuadas durante o ciclo de vida do projeto. Esta área, chamada de **estimativas de esforço**, conta com algumas técnicas que têm apresentado resultados interessantes nos últimos anos.

Técnicas de estimativas

As técnicas de estimativa de *software* são utilizadas para se chegar às estimativas de custo, de produtividade, de tempo etc. Elas são fundamentais para que se obtenha sucesso em um projeto de desenvolvimento de *software*.

Ao longo do tempo, foram sendo propostas diversas técnicas para apoiar a atividade de estimativas. A seguir, confira alguns exemplos.

- Pontos de histórias.
- Pontos de função.
- Pontos por caso de uso.
- Constructive Cost Model (COCOMO).
- Delphi.

A maioria das técnicas de estimativas utilizam pelo menos um parâmetro como base, por isso são chamadas **técnicas paramétricas**. No entanto, há técnicas que não usam nenhum parâmetro, chamadas **não paramétricas**. Um exemplo é quando a estimativa é utilizada por um especialista. Neste caso, simplesmente pede-se a pessoas com bastante experiência em desenvolvimento de *software* que deem sua opinião sobre o tempo e esforço total de desenvolvimento.

Algumas técnicas se baseiam no conjunto de requisitos identificados para o projeto, como: pontos de histórias, pontos de função e pontos de caso de uso. Ao longo das próximas seções e Unidades, conheceremos a base e o funcionamento delas.

Pontos de histórias (*story points*)

Pontos de histórias é a estimativa preferida dos métodos ágeis. A técnica se aproxima muita da estimativa de especialista, uma vez que a equipe ágil usará hipóteses para estimar o esforço de cada história de usuário. A técnica tem a vantagem de poder ser aplicada a praticamente qualquer tipo de sistema, quantos pontos de função se aplicam mais facilmente a sistemas comerciais. Sistemas como jogos, compiladores ou aplicações matemáticas poderão apresentar grandes dificuldades para a aplicação de pontos de função.

Uma técnica bastante utilizada usando pontos de histórias é o **planning poker**. Nela, o líder do projeto prioriza o que deverá ser realizado primeiro, mas o time de desenvolvimento é quem decide quanto tempo levará o desenvolvimento. Há muita colaboração entre os times, o que aumenta o comprometimento. As estimativas são ajustadas ao longo do processo.

Para aplicação da técnica, é utilizado um baralho (Figura 3) e a série de Fibonacci customizada (0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?) para estimar.



SAIBA MAIS

Para entender o detalhe do funcionamento da técnica, consulte esse material:

<https://www.luiztools.com.br/post/planning-poker-como-estimar-tempo-de-desenvolvimento-de-software/>

Figura 3: *Planning poker*



Técnica que utiliza histórias de usuários e a escala de Fibonacci para estimativas de software. Fonte: Autor (2021).



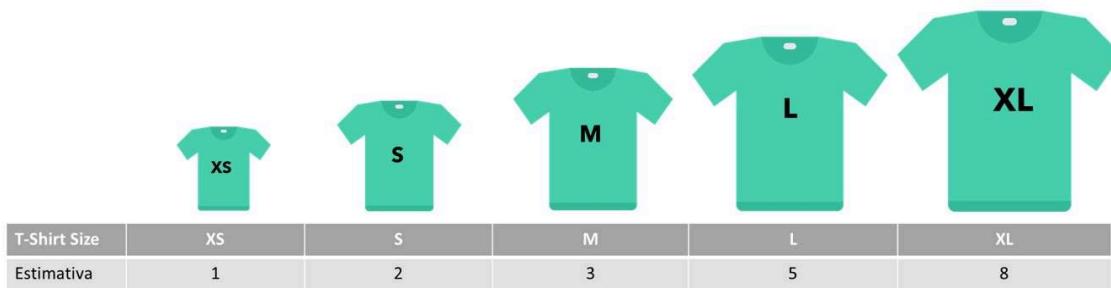
SAIBA MAIS

A técnica pode ser aplicada com um baralho físico ou ainda podem ser utilizadas ferramentas virtuais para a sua aplicação. Segue o link de uma dessas ferramentas.

<https://www.planitpoker.com/>

Nos métodos ágeis, a importância da estimativa normalmente está na comparação entre histórias, ou seja, mais importante do que saber quantos dias uma história efetivamente levaria para ser implementada, é saber que uma história levaria duas

vezes mais tempo do que outra para ser implementada.



Além do *planning poker*, há outras técnicas que usam pontos por histórias, uma delas é a camiseta, ou *t-shirt sizes*. Nesta técnica, utilizam-se valores, tais como “pequeno”, médio” e “grande”. Ficou curioso sobre o seu funcionamento? Realize uma busca pela internet!

As técnicas e as ferramentas evoluem o tempo todo, precisamos desenvolver a capacidade de pesquisar e de senso crítico para identificar quais são os instrumentos mais eficientes para o contexto que está sendo vivenciado.

Nesta Unidade, houve uma introdução às atividades de medição e apresentação do quanto elas são importantes para a qualidade do *software* e o planejamento do projeto. Em seguida, aprofundamos o entendimento da atividade de estimativa de esforço, que é essencial para estimar o custo e o tempo de desenvolvimento de software. Existem várias técnicas disponíveis para apoiá-la atividade, desde técnicas informais, que contam com a experiência de um especialista, conhecidas como técnicas não paramétricas, até técnicas baseadas em parâmetros, como os requisitos identificados no *software* a ser desenvolvido. As técnicas baseadas em parâmetros são conhecidas como paramétricas.

| Referências bibliográficas

DUARTE, L. Planning poker: como estimar tempo de desenvolvimento de software. **Luiz Tools**, Agile, 2016. Disponível em: <https://www.luiztools.com.br/post/planning-poker-como-estimar-tempo-de-desenvolvimento-de-software/>. Acesso em: 26 abr. 2021.

GOMES, A. E. Métricas e estimativas de software - o início de um rally de regularidade. **Linha de Código**, Gerência, Estimativas, 2021. Disponível em: <http://www.linhadecodigo.com.br/artigo/102/metricas-e-estimativas-de-software-o-inicio-de-um-rally-de-regularidade.aspx>. Acesso em: 26 abr. 2021.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788580555349/>.

WAZLAWICK, R. S. **Engenharia de software**: conceitos e práticas. 2. ed. Rio de Janeiro: Elsevier, 2019.

VIANA, J. Planning poker e pontuação de fibonacci. **Agile Pink**, 2019. Disponível em: <https://agilepink.com/planning-poker/2019/>. Acesso em: 26 abr. 2021.



© PUCPR - Todos os direitos reservados.