Fronteiras da Engenharia de Software - Podcast

Episódio 2, Temporada 2: Estimativa de Software, com Emilia Mendes (BTH)

Host: Adolfo Neto (UTFPR)

Co-host: Ingrid Nunes (UFRGS)

Equipe: Danilo Ribeiro (Zup), Leonardo Fernandes (IFAL), Fabio Petrillo (Univ. Quebec), Gustavo Pinto (UFPA)

| Título: Estimativas de Software com Emilia Mendes (Blekinge Institute of Technology) |
| --- |
| Resumo:  Conversamos com Emilia Mendes, professora do Blekinge Institute of Technology, na Suécia, sobre estimativas de software.  Sites da Emilia:   * <https://bit.ly/EmiliaMendes> * Scholar <https://scholar.google.com/citations?user=Mz4IbXgAAAAJ&hl=en>   Nossa equipe é:   * Leonardo Fernandes (IFAL) * Gustavo Pinto (UFPA) * Fabio Petrillo (UQAC) * Danilo Monteiro (ZUP / ITAÚ) * Adolfo Gustavo Serra Seca Neto (UTFPR) * Ingrid Nunes (UFRGS)   Nosso site é <https://fronteirases.github.io/>.  A edição deste episódio foi feita pela Zup Innovation. |

# Script do Episódio

## Parte 1: Apresentação [5 min, estimativa]

[ADOLFO] Hoje no Fronteiras da Engenharia de Software vamos conversar com a Emilia Mendes, professora no Blekinge Institute of Technology, na Suécia. Nós vamos falar sobre estimativas de software.

[ADOLFO] Tudo bem, Emília? Você pode se apresentar para as pessoas que nos escutam?

## Parte 2 - Estimativas de Software [20 min, estimativa]

[INGRID] Quando a gente fala em estimativas de software, a que estamos nos referindo? Vamos estimar o que mais precisamente?

[ADOLFO] Com relação ao tamanho, quais as diferentes medidas que existem? Existe alguma que seja melhor do que a outra? Linha de código é, por exemplo, uma medida de tamanho adequada?

[INGRID] Então temos três dimensões a serem estimadas, tamanho, esforço (por exemplo em man-month), e cronograma. Como essas estimativas se relacionam? A gente estima uma para derivar outras?

[ADOLFO] Para fazer essas diferentes estimativas, que técnicas existem? Quais fatores devem ser considerados por impactarem nas estimativas, por exemplo a experiência do desenvolvedor, o segmento de mercado da empresa, etc.?

[INGRID] E como aparecem os requisitos nesse contexto? Eles precisam estar definidos precisamente?

[ADOLFO] Tempos atrás houve a proposta de muitos modelos focados na estimativa de esforço. Foi neste contexto que surgiu, por exemplo, o COCOMO II. Hoje vemos menos trabalhos sobre isso. Você tem uma visão do porquê disso?

[INGRID] Hoje em dia, a maioria dos projetos segue uma abordagem ágil de desenvolvimento. Essas técnicas que existem focam em processos de desenvolvimento de software tradicional ou são aplicáveis em projetos que seguem métodos ágeis também?

[ADOLFO] Nos projetos que adotam métodos ágeis, o planning poker é uma forma comum de estimativa de tamanho (ou até de esforço) usando story points. Você pode explicar como ele funciona? E, já aproveitando, como se derivam as outras medidas, como tempo, a partir disso?

[INGRID] Planning poker é então uma variação da técnica de estimativa wideband delphi? Quais as diferenças entre elas?

[ADOLFO] Basicamente, as estimativas se baseiam na experiência pessoal do desenvolvedor. Isso é confiável? Ou os projetos deveriam ter histórico armazenado para fazer estimativas de forma objetiva?

[INGRID] Esse problema de estimar tamanho ou esforço de software parece ainda não ser resolvido. Na sua opinião, por que isso acontece? O que falta? Isso será resolvido algum dia, como ocorre em outros produtos feitos por engenharia?

## Parte 3: Pesquisa sobre Estimativa de Esforço [3 min, estimativa]

~~[?] Essa questão de usar dados históricos de projetos é abordada no seu artigo "Transfer learning in effort estimation". Você poderia por favor explicar o que seria a transferência de aprendizado?~~

[ADOLFO] Em alguns dos seus trabalhos você fala de estimativa de esforço no contexto web. Por que é necessário olhar para o esse contexto de forma separada de outros contextos?

## Parte 4: Próxima Fronteira da Engenharia de Software [3 min, estimativa]

[INGRID] Emilia, falando um pouco sobre você, embora você tenha feito parte da sua formação no Brasil, você fez sua carreira no exterior, passando pela Inglaterra, Nova Zelândia, e agora na Suécia. É possível comparar a carreira do pesquisador de engenharia de software nesses diferentes países? Em quais aspectos existe diferença na carreira no exterior e no Brasil?

[ADOLFO] De que forma esses diferentes países tratam a relação entre pesquisa e prática de desenvolvimento de software? Há algum que fornece mais incentivos doque os outros?

## Parte 5: Próxima Fronteira da Eng [3 min, estimativa]

[ADOLFO] Emilia, para você, qual é a próxima fronteira da engenharia de software? (pode ser algo que você acha que vai acontecer ou que você gostaria que acontecesse em nossa área)

## Parte 6: Encerramento

Adolfo agradece e passa para Emilia.

Ingrid fecha o episódio.

# Texto para divulgação

Conversamos com Emilia Mendes, professora do Blekinge Institute of Technology, sobre Estimativa de Software.

Sites de Emilia

* <https://scholar.google.com/citations?user=Mz4IbXgAAAAJ&hl=en>
* <https://www.bth.se/eng/staff/emilia-mendes-eme/>

Links Citados

# Perguntas Antigas

[] Estimar esforço é uma atividade extremamente importante e recorrente no desenvolvimento de software (desenvolvedores precisam estimar suas atividades de cada sprint), mas mesmo assim ainda é muito difícil fazer uma estimativa correta, ou até mesmo uma estimativa próxima. Como resultado, não é comum termos sprints com entregas atrasadas ou com funcionalidades pendentes. Além disso, também não é incomum os desenvolvedores discordarem no quanto de esforço que seria necessário pra realizar uma dada atividade (basta acompanhar alguma seção de planning poker). Por que ainda é tão difícil estimar precisamente o esforço para uma atividade um pouco longe do trivial?

[] Precisamos estimar esforço sempre? Ou há tarefas que, de tão corriqueiras que são, nem precisamos mais estimar esforço? Existe algum cenário em que estimar esforço é burocrático, ou até mesmo desnecessário?

[] A experiência do desenvolvedor sem dúvida pode ajudar na estimativa de uma atividade. Mas além da experiência, o que mais pode ajudar?

[] E o contrário? O que pode dificultar a estimativa de esforço? Requisitos não bem definidos parem ser um complicador para estimar esforço adequadamente, correto? O que mais complica?

[] Estimar esforço é uma atividade que vai na contramão de práticas ágeis? Por exemplo, um dos pilares do manifesto ágil é que “Software em funcionamento mais que documentação abrangente”, mas se não tivermos uma documentação relativamente abrangente, podemos dificultar a estimativa de esforço (pois os desenvolvedores podem não saber exatamente o que precisa ser feito), que por sua vez, poderia impactar negativamente no software funcionando. Faz sentido?

[] Além do planning poker que é bem conhecido no Scrum, quais outras práticas que desenvolvedores utilizam para estimar esforço? Dessas diferentes práticas, há alguma mais adequada? Ou vai sempre depender do contexto?