**Fronteiras da Engenharia de Software - Podcast**

Host: Adolfo Neto (UTFPR)

Co-host: Ingrid Nunes (UFRGS)

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

**Introdução ao Episódio**

**CORTAR ATÉ 9:42**

[ADOLFO] Olá ouvintes, eu sou Adolfo Neto, professor da Universidade Tecnológica Federal do Paraná.

Hoje começamos a segunda temporada do podcast Fronteiras da Engenharia de Software.

Muito obrigado a todos e todas que nos escutam desde nosso primeiro episódio.

Em primeiro lugar, dou as boas-vindas a Ingrid Nunes, que foi entrevistada no episódio 8 e agora é parte da nossa equipe e vai atuar como apresentadora junto comigo hoje.

Tudo bem, Ingrid?

[INGRID] Oi pessoal, tudo bem?

[ADOLFO]

Tudo bem, Ingrid.

Seja bem-vinda.

O Fronteiras da Engenharia de Software é um podcast de divulgação científica em Engenharia de Software.

Nossos episódios abordam desde questões fundamentais da engenharia de software até os mais recentes avanços na área.

[ADOLFO] Neste episódio, conversaremos com Fernando Castor, professor da UFPE (Universidade Federal de Pernambuco) em Recife, sobre compreensão de código.

**Episódio 1, Temporada 2: Compreensão de código, com Fernando Castor (UFPE)**

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

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

**Parte 2 - Compreensão [20 min, estimativa]**

[ADOLFO] Castor, quem já programou e teve que entender código feito por outros sabe que tem código que é bem difícil de compreender pela forma como ele foi escrito. Por exemplo, podem haver variáveis, funções ou métodos com nomes poucos intuitivos, ou listas de parâmetros muito grandes, o que pode fazer com que seja difícil usar a função ou método.

Só que esta noção de compreender código é subjetiva, não?

Existe uma definição mais precisa sobre o que seja **compreensão de código**?

**CORTAR de 13:52 a 14:45**

[INGRID] O código pode ser difícil de entender porque ele implementa algo complexo como também porque não foi bem escrito. Quando falamos que um código é de difícil compreensão, pode ser em função de qualquer um desses dois motivos?

[ADOLFO] Normalmente se diz que código que é difícil de entender acaba por ser mais difícil de evoluir, porque antes de mexer no código o desenvolvedor tem que entender ele. Também, pelo baixo entendimento do código, pode ser que bugs sejam introduzidos mais facilmente. Isso é apenas senso comum ou existem evidências concretas de benefícios relacionados com código que tenha uma alta compreensão?

[INGRID] E falando mais a fundo dessa relação entre código difícil de entender e difícil de evoluir, um não existe sem o outro? Eles são a mesma coisa? Ou é possível apontar alguma linha que separa essas duas características do código?

[ADOLFO] E como podemos escrever código de fácil compreensão? Existem as práticas divulgadas em *Clean Code (Código Limpo)*, que são relativamente bem exploradas e conhecidas, como o *DRY* (*don't repeat yourself* - Não se Repita) e *Keep it simple* (Mantenha Simples). Que outros guias podem ajudar?

[INGRID] Desenvolvedores defensores de Python falam que código nessa linguagem é de certa forma mais fácil de entender. Assim, a linguagem de programação escolhida tem algum impacto na compreensão de código?

35:20 a ---

13h:47min talvez regravar esta pergunta depois - se escutando ficar muito ruim

(precisa mesmo

**Parte 3: Pesquisa sobre compreensão de código [3 min, estimativa]**

[ADOLFO] Tem um artigo de Scalabrino e outros, onde eles avaliaram o nível de compreensão de código de acordo com desenvolvedores e correlacionaram essa medida com 121 métricas de código. Nenhuma delas tem correlação. Eles argumentam que deve uma haver uma nova métrica que capture isso. Na sua opinião, por que você acha que isso acontece? ~~Ou você acha que um código bem estruturado (de acordo com métricas de modularidade, por exemplo) não é necessariamente fácil de entender?~~

Corte de 49:25 a 50:03

**~~14h:00min CORTE AQUI - teste do microfone~~**

[INGRID] Atualmente, existem vários trabalhos que mineram repositórios de código para construir modelos de aprendizado e fazer recomendações para os desenvolvedores. Um exemplo é recomendadores de nomes de métodos, algo relacionado com a compreensão de código. Uma discussão a respeito disso é que existe a suposição de que os nomes dos métodos de onde se construindo o modelo de aprendizado são bons nomes. Qual sua opinião a respeito? Não faria mais sentido fazer recomendações com base em boas práticas?

[ADOLFO] Você tem trabalhado recentemente sobre compreensão de código em Swift. Primeiro, por que é importante estudar compreensão de código em Swift? Você acha que a mistura de paradigmas de programação poderia dificultar a compreensão de programas em Swift?

[ADOLFO] Você pode agora explicar um pouco mais sobre o trabalho e o que vocês observaram?

[INGRID] Além desse trabalho sobre swift, você tem alguns trabalhos sobre confusão em revisões de código. Você poderia falar um pouco sobre esses trabalhos, e qual a relação deles na compreensão de código?

**Parte 4: Próxima Fronteira da Eng [3 min, estimativa]**

[ADOLFO] Fernando, para você, qual é a próxima fronteira da engenharia de software?

**Parte 6: Encerramento**

ADOLFO agradece e passa para Castor. Ingrid fecha o episódio.

**Texto para divulgação**

Conversamos com Fernando Castor, professor da UFPE, sobre compreensão de código.

Sites de Castor

<http://cin.ufpe.br/~fjclf>

<https://sites.google.com/a/cin.ufpe.br/castor/>

<http://fernandocastor.github.io/>

<https://twitter.com/fernandocastor>

<https://github.com/fernandocastor>

Links Citados

* Identifying Confusing Code in Swift Programs <https://docs.google.com/a/cin.ufpe.br/viewer?a=v&pid=sites&srcid=Y2luLnVmcGUuYnJ8Y2FzdG9yfGd4OjY2YTBhNWFjZWYxNmRiMTA>
* How Swift Developers Handle Errors <https://www.win.tue.nl/~aserebre/MSR2018-Nathan.pdf>
* The Mind Is a Powerful Place: How Showing Code Comprehensibility Metrics Influences Code Understanding (sobre Anchoring Effect) <https://www.computer.org/csdl/proceedings-article/icse/2021/029600a512/1sEXo9o0BeE>, <https://twitter.com/MarvinWyrich/status/1339906400764497920?s=20>
* Vídeo do artigo acima: <https://youtu.be/N_lpaCqnE3s>