**Fronteiras da Engenharia de Software - Podcast**

Host: Adolfo Neto (UTFPR)

Co-host: Gustavo Pinto (UFPA)

Equipe: Danilo Ribeiro (Zup), Leonardo Fernandes (IFAL), Fabio Petrillo (Univ. Quebec), Ingrid Nunes (UFRGS)

**Introdução ao Episódio**

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

[GUSTAVO] Oi pessoal, tudo bem? Eu sou Gustavo Pinto, professor da UFPA, e este é mais um episódio do Fronteiras da Engenharia de Software, um podcast feito para refletir sobre o presente e futuro da engenharia de software

[ADOLFO] Fronteiras da Engenharia de Software é um podcast de divulgação científica que discute mensalmente diversificados temas da Engenharia de Software. Os episódios abordam desde questões fundamentais da engenharia de software até os mais recentes avanços na área.

[GUSTAVO] Nosso convidado deste décimo segundo episódio é o prof Paulo Borba. Paulo fez doutorado em Oxford e hoje é professor titular no Centro de Informática da UFPE. Paulo investiga e desenvolve ferramentas e técnicas para melhorar os níveis de produtividade e qualidade de desenvolvimento de software, especialmente reduzindo o esforço e frustração. No passado, Paulo foi também dos fundadores da empresa Qualiti, no Porto Digital.

[ADOLFO] Neste episódio, conversaremos com Paulo sobre modularidade de software.

**Episódio 12: Modularidade de código, com Paulo Borba (UFPE)**

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

[ADOLFO] Olá ouvintes, para este décimo segundo episódio, contamos com a presença de Paulo Borba, que é professor na UFPE (Universidade Federal de Pernambuco) em Recife.

Tudo bem, Paulo? Você pode se apresentar para as pessoas que nos escutam?

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

[GUSTAVO] Modularização é um conceito que vem desde os princípios da programação. Por exemplo, na década de 70, o David Parnas já escrevia sobre critérios para decompor sistemas de software em módulos (que por sinal é o seu artigo mais citado, com 7000 citações). Mas, o que seria modularizar um código? Quebrar em funções ou classes menores é modularização?

[ADOLFO] Paulo, e o que **não** seria modularização? Existe alguma crença que desenvolvedores de software tem sobre modularização, que eles acham que estão fazendo, quando na verdade não estão? Ou então, quais são as más práticas de modularização de software?

[GUSTAVO] No artigo do David Parnas, ele começa dizendo que, abre aspas “modularização é um mecanismo para melhorar a flexibilidade e compreensão de um sistema, de forma a diminuir o seu tempo de desenvolvimento”. Mas o que ele quis dizer com isso? De que forma modularizar um sistema pode impactar na flexibilidade e compreensão de código?

[GUSTAVO] Qual a relação de modularização e os princípios de design de software? Por exemplo, alguns dos princípios de SOLID parecem ter alguma relação com modularidade, como o princípio de responsabilidade única (em que o módulo só pode ter uma única razão para mudar) e o princípio aberto/fechado (em que uma classe pode ser aberta para extensão mas fechada para modificação). Ter um bom design de software implica que eu tenho um código bem modularizado?

[ADOLFO] Embora conceitos como polimorfismo e information hiding pudessem ser exercitados em uma linguagem de programação como C, linguagens orientadas a objetos tornaram seu uso mais seguro (pois o programador não precisa mais lidar com ponteiros) e conveniente. Mas quais os avanços mais recentes? De que forma as linguagens de programação tem avançado de forma a garantir novos mecanismos de modularização de software?

// pequenos problemas de áudio do Paulo aqui

[ADOLFO] Paulo, as técnicas de modularização são específicas de paradigma de programação? Com a recente popularização do paradigma de programação funcional, os princípios de modularização mudaram? São os mesmos?

// fazer um pequeno corte aqui no final

[GUSTAVO] Paulo, apesar de muitos dos princípios de modularização serem bem conhecidos e até bem utilizados, ainda temos hoje uma dificuldade muito grande em criar sistemas bem modularizados. Pra dar um exemplo, basta pedir pro time de desenvolvimento trocar de framework, por exemplo, mude de Angular para Vue. Até mesmo mudar de versões do mesmo framework é muitas vezes complicado (o pessoal do Rails sabe a dor que era fazer atualizações das versões 3, 4, 5, etc). Na sua opinião, ao que se deve essa dificuldade de criar sistemas, de fato, fracamente acoplados?

// coloquei uma pergunta nova, e tem que cortar um pedaço aqui.

Como a gente coloca essa visão de médio/longo prazo dentro do orçamento do projeto? Dentro de uma sprint de entrega.

**Parte 3: Pesquisa sobre modularidade [10 min, estimativa]**

[ADOLFO] Paulo, você tem trabalhado mais recentemente com o problema de conflitos de merge. Você pode explicar o que seria esse problema?

//pergunta refeita

<AGUARDAR>

[ADOLFO] ~~A primeira vista parece que conflitos de merge não tem muita relação com modularidade, mas estou enganado, né? Qual é a relação de conflitos de merge e modularidade?~~ Se os desenvolvedores fossem mais disciplinados com a modularidade dos seus sistemas, eles teriam menos problemas de merge? Práticas de integração contínua não deveriam diminuir esses problemas?

[GUSTAVO] Quais são os diferentes tipos de problemas de merge. Por exemplo, tem o merge estruturado, o semi-estruturado, o semântico. Esses diferentes problemas de merge podem impactar de forma diferente na modularidade de um sistema?

(CoRTE AQUI)

[GUSTAVO] Paulo, vc tem um artigo recente chamado "The Impact of Structure on Software Merging: Semistructured versus Structured Merge.” Você pode falar um pouco mais sobre esse trabalho?

**Parte 4: [Abordar outro tema de interesse] [5 min, estimativa]**

[ADOLFO] Paulo, vc tem investido recentemente na criação de ferramentas que dão apoio a estas atividades de resolução de conflitos de merge. Qual é a sua visão com relação a criação dessas ferramentas? Fortalecer a relação com a prática?

CORTE AQUI (aliança)

[GUSTAVO] Você está há várias décadas envolvido com a nossa comunidade de engenharia de software no Brasil. A gente nem precisa comentar sobre a situação atual do país. Mas mesmo assim parece que a comunidade parece que tem se fortalecido. Você tem alguma intuição do porque desse progresso?

Um cortezinho no final da fala do Paulo

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

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

**Parte 6: Encerramento**

ADOLFO agradece e passa para Paulo. Gustavo fecha o episódio.

**Texto para divulgação**

Conversamos com Paulo Borba, professor da UFPE, sobre Modularidade de Software.

Sites de Paulo

* <https://dblp.org/pid/b/PauloBorba.html>
* <https://scholar.google.com.br/citations?user=PpRDLgcAAAAJ&hl=de>
* <https://pauloborba.github.io/>

Links Citados

* <https://www.win.tue.nl/~wstomv/edu/2ip30/references/criteria_for_modularization.pdf>