

Maturidade no processo de desenvolvimento de software

Qualidade de código contínua

João Carlos Brasileiro Stefenon de Almeida¹

¹ Universidade do Vale do Rio dos Sinos (UNISINOS)
São Leopoldo – RS – Brazil

jcb Brasileiro@hotmail.com

Abstract. *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

Resumo. *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

Part I

INTRODUÇÃO

Uma das mudanças mais impactantes no processo de desenvolvimento de software das duas últimas décadas, foi a concepção e absorção das metodologias ágeis de software, seja SCRUM, XP ou outra, todas foram criadas a partir da filosofia AGILE [Agile 2001], surgido a partir do esforço de várias pessoas que lidavam com o processo de software na década de 1990, visando uma abordagem mais efetiva e eficiente para o desenvolvimento de software.

1º e 2º parágrafo: apresentação do tema dentro de um contexto.

3º parágrafo: delimitação do tema, apresentado através do problema de pesquisa do seu TCC.

4º parágrafo: apresente possíveis respostas para o problema de pesquisa levantado, ou seja, as hipóteses.

5º parágrafo: em poucas palavras, fale sobre o objetivo geral do trabalho e também dos específicos. Eles são ingredientes fundamentais para o trabalho.

6º parágrafo: apresente a relevância do seu trabalho acadêmico, identificando a

importância dele para a sociedade ou comunidade científica. Isso é o que chamamos de justificativa.

7º parágrafo: descreva, em poucas palavras, qual metodologia foi utilizada. Foi pesquisa bibliográfica ou de campo? Você deve especificar o procedimento de forma concisa.

8º parágrafo: apresente a estrutura do trabalho, ou seja, como ele está dividido em capítulos. Lembre-se de falar, resumidamente, sobre o que se trata cada capítulo.

Part II

REVISÃO BIBLIOGRÁFICA

1. *Object-oriented programming (OOP)*

2. *Clean Code*

2.1. *Object-Oriented Design Principles (OODP)*

2.2. *General responsibility assignment software principles (GRASP)*

[https://en.wikipedia.org/wiki/GRASP_\(object-oriented_design\)](https://en.wikipedia.org/wiki/GRASP_(object-oriented_design))

- Controller
- Creator
- Indirection
- Information Expert
- High Cohesion
- Low Coupling
- Polymorphism
- Protected Variations
- Pure Fabrication

2.3. *S.O.L.I.D principles*

- Single responsibility principle
- Open closed principle
- Liskov substitution principle
- Interface segregation principle
- Dependency Inversion principle

3. *Test-driven development (TDD)*

Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle: first the developer writes an (initially failing) automated test case that defines a desired improvement or new function, then produces the minimum amount of code to pass that test, and finally refactors the new code to acceptable standards.

The process can be defined as such:

Write a failing unit test Make the unit test pass Refactor

Repeat this process for every feature, as is necessary.

4. Trabalhos Relacionados

References

Agile (2001). Manifesto for agile software development.
<http://agilemanifesto.org/>.