CENTRO UNIVERSITÁRIO DE PATOS - UNIFIP
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PESQUISA - COOPIC
COORDENAÇÃO DE EXTENSÃO - COOEX
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CODE DESIGN: Estilo e Compreensão de Código

Orientador: Profº. Me. Pablo Roberto Fernandes de Olvieira

Alunos-Pesquisadores: Antonio Lucas Araujo de Medeiros Calixto; Antônio Lucas Cavalcante Félix

RESUMO

O Design de Software é uma das etapas mais importantes do ciclo de vida do desenvolvimento de Software. Nesta fase faz-se a Análise e o Projeto de um Sistema, define-se a Arquitetura, tecnologias a serem utilizadas, etc. É importante que nesta etapa de desenvolvimento, o projeto seja claro, limpo e o menos complexo, com vista em sua implementação/codificação. Desta forma, boas práticas de programação, padrões de projetos, estilos de codificação e design de código devem ser seguidos para melhorar a compreensão e legibilidade deste código, tendo em vista que lemos código muito mais do que escrevemos. Logo, é importante adotar estilos de escrita que melhorem o processo de leitura e compreensão do código.

INTRODUÇÃO

A adoção e adaptação de determinados estilos de código desempenham um papel fundamental no desenvolvimento de software, fornecendo diretrizes para formatação, nomenclatura e estrutura do código-fonte, promovendo legibilidade, coesão e facilitando a manutenção (Broekhuis, 2021).

Em um estudo conduzido por Santos e Gerosa (2018), que investigaram por meio de um Survey a influência de estilos de codificação em Java na legibilidade percebida por desenvolvedores de software, alguns estilos revelaram melhorias na legibilidade, enquanto outros não demonstraram efeitos estatisticamente significativos. Um dos estilos analisados, chegou a comprometer a legibilidade, entendida como a qualidade relacionada à clareza, compreensão e facilidade de interpretação de um código-fonte por outros desenvolvedores.

Corroborando com os trabalhos anteriores, observa-se que pequenos padrões indicados em guias de estilo também podem impactar negativamente na legibilidade de um código. Bem como, a diversidade de linguagens de programação impede a existência de um conjunto universal de regras de estilo, sendo que cada linguagem possui suas próprias definições específicas (Allamanis et al., 2014).

Deste modo, estudos empíricos são necessários para avaliar pequenos padrões de código, pois os mesmos podem impactar a legibilidade e afetar a compreensão do desenvolvedor.

Observa-se na Figura 1 que o estilo de codificação da direita é diferente da esquerda, embora ambos os trechos de código possuam um mesmo resultado. Em Costa et al. (2023) esses dois trechos foram comparados, e a entrevista com desenvolvedores novatos revelaram que o padrão da direita demonstrou ser mais legível para a maioria.

```
if (not cont == 3):     if (cont != 3):
```

Figura 1. Pequenos padrões de código (COSTA et al. 2023)

OBJETIVOS

Identificar pequenos padrões de código que podem afetar a legibilidade

Verificar a presença de átomos de confusão em padrões de guias de estilo

Aplicar um Survey com desenvolvedores novatos comparando estilos de código

Verificar a preferência de estilos de código por desenvolvedores novatos

METODOLOGIA

Nossa pesquisa segue a abordagem Goal/Question/Metric (GQM) para nortear o processo da escolha das métricas que seriam medidas (Basili et al., 1994). Comparamos programas contendo padrões presentes em guias de estilo, com versões compatíveis e incompatíveis, segundo as mesmas, funcionalmente equivalentes, com o objetivo de avaliar como o método Survey pode ser usado para obter novas percepções com relação à legibilidade de código do ponto de vista de novatos em linguagens de programação, no contexto de tarefas adaptadas conforme as sugestões de guias de estilo.

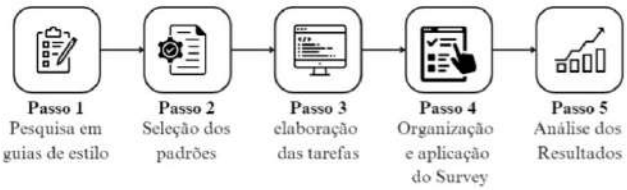


Figura 2. Passos seguidos na pesquisa

VALIDAÇÃO

Os resultados obtidos por meio deste trabalho refletirão em direcionamentos importantes para a criação e manutenção de guias de estilo, bem como sua adoção por parte de empresas e professores. Uma vez que os resultados apontam quais padrões são melhores lidos e compreendidos por parte de desenvolvedores novatos, ou seja, que ainda estão em fase de aprendizagem e adaptação à determinada linguagem. Espera-se que esse estudo possa ser continuado para investigar outras padrões de códigos e verificar estruturas code smells, syntax sugar ou átomos de confusão.

REFERÊNCIAS

Broekhuis, S. The importance of coding styles with in industries. 2021. Trabalho de Conclusão de Curso. University of Twente

Santos, R. M. a. dos; Gerosa, M. A. Impacts of Coding Practices on Readability. In: Proceedings of the 26th Conference on Program Comprehension (ICPC '18). New York, NY, USA: ACM, 2018. p. 277–285. ISBN 978-1-4503-5714-2.

Allamanis, M; Barr, E. T; Bird, C. and Sutton, C. "Learning natural coding conventions," in Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE), 2014, pp. 281–293.

Costa, J. A. S.; Gheyi, R.; Castor, F. Oliveira, P. R. F; Ribeiro, M. and Fonseca, B. Seeing confusion through a new lens: on the impact of atoms of confusion on novices' code comprehension. Empirical Software Engineering, 28(4):81, 2023.

Basili V, Caldiera G, Rombach H (1994) The Goal Question Metric Approach. Encycl Softw Eng 2:528–532