INSTITUTO FEDERAL
SUDESTE DE MINAS GERAIS | Campus Juiz de Fora

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

UNNECESSARY COMPLEXITY REMOVAL

This briefing reports scientific evidence on the use of an automated refactoring approach to remove unnecessary complexity in source code.

FINDINGS

- The findings presented in this briefing are from an experimental study conducted to evaluate the functionalities of a tool (complexity tool) that measures cyclomatic complexity from source code, and suggests a refactored version of it to eliminate unnecessary complexity in its structure.
- It is believed that less cyclomatic complexity equals less effort to develop unit tests.
- In a previous study, 482 students' programming assignments were analyzed. 16% of them had unnecessary conditional expressions and could be improved in terms of their cyclomatic complexity.
- A study was conducted with subjects to evaluate how complex (cyclomatic complexity) was their source code and how long (in minutes) they would take to develop unit tests.
- Subjects were Information Systems students from 4th and 3rd year, whose results are shown in Figures 1 and 2 respectively.

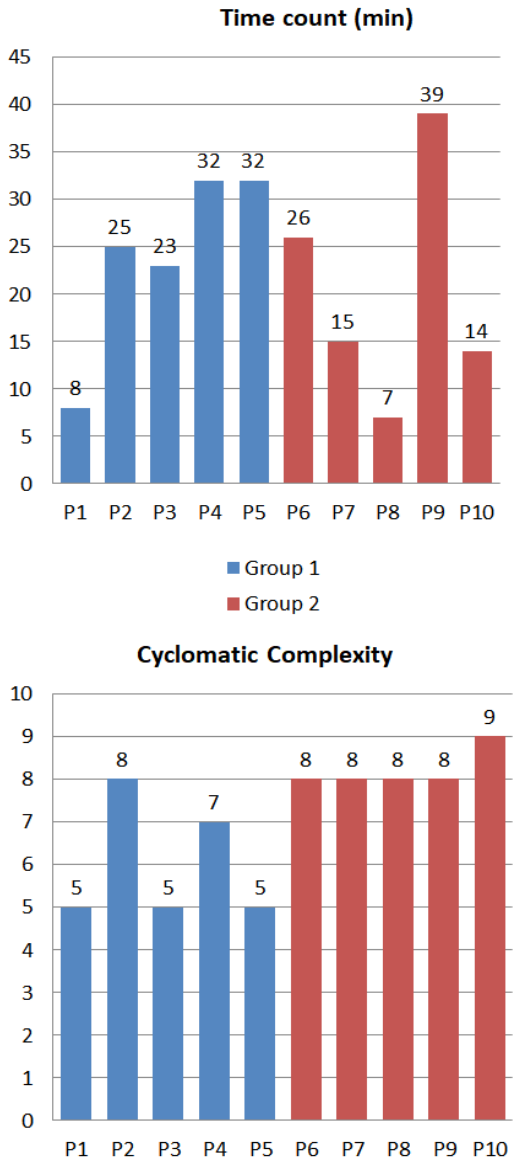

Figure 1: Graphical results from experimental study applied to the 4th year students.

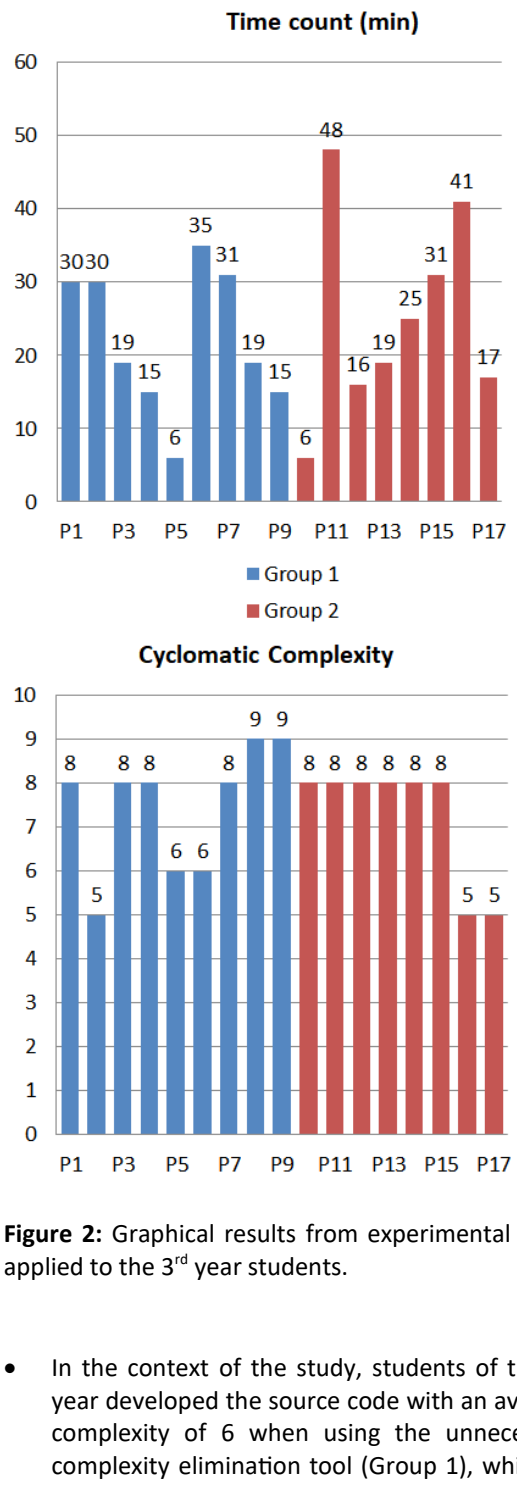

Figure 2: Graphical results from experimental study applied to the 3rd year students.

- In the context of the study, students of the 4th year developed the source code with an average complexity of 6 when using the unnecessary complexity elimination tool (Group 1), while for those who did not use, they had an average complexity of 8 (Group 2).
- They took, on average, 24 minutes to develop unit test cases when using the complexity tool (Group 1), while those who did not use took about 20 minutes (Group 2), on average.
- Students of the 3rd year developed the source code with an average complexity of approximately 7 when using and not using the unnecessary complexity elimination tool (Groups 1 and 2).
- They took, on average, approximately 22 minutes to develop unit test cases when using the complexity tool (Group 1), and approximately 25 minutes, on average, for those who did not use (Group 2).
- The tool successfully helped 4th year students in developing source code without unnecessary cyclomatic complexity.
- However, it was not possible to confirm that the tool helped decreasing unit tests develop time.

Keywords

Cyclomatic Complexity
Software Quality
Source Code Refactoring
Software Testing
Control Flow Graph

Who is this briefing for?

Software engineering practitioners who want to use a tool to remove unnecessary cyclomatic complexity in source code.

Where the findings come from?

All findings of this briefing were extracted from an experimental study conducted by Magalhães et al.

What is included in this briefing?

Results obtained in the experimental study.

What is not included in this briefing?

Statistical analysis about the experimental study and functionalities description of complexity tool are both not included in this evidence briefing.

ORIGINAL RESEARCH REFERENCE

Magalhães, N. M.; Campos Junior, H. S.; Araújo, M. A. P.; Neves, V. O.; ***An Automated Refactoring Approach to Remove Unnecessary Complexity in Source Code.*** Proceedings of the 2nd Brazilian Symposium on Systematic and Automated Software Testing. 2017. https://doi.org/10.1145/3128473.3128476