Investigating the correlation between prior programming experience and cyclomatic code complexity in student software projects

Collaboration in Computer Science Education

Supervisors:
Ir. Merel Steenbergen,
Dr.ir. E. Fenia Aivaloglou

Examiner:
Dr.ir. Mitchell Olsthoorn

Author: Beatriz Barroso
beatrizbarroso@tudelft.nl

1. Background

Assessing student contributions in SE projects remains a challenge, as there are various metrics that can be relied upon [1]. Cyclomatic complexity often serves as a proxy for code quality [2]. While it has been previously studied within education, its relationship with prior programming experience remains unclear. Additionally, prior programming experience's effect on code quality has been extensively researched [3], however few studies have analyzed the impact of prior programming experience on structural quality within collaborative projects. Lastly, prior experience has been linked to years of study within academia, with no correlation found with professional experience [4].

Aim: Analysis of correlations to understand if prior programming experience plays a role on cyclomatic complexity scores in student projects.

2. Research Questions

Is there a correlation between prior programming experience and cyclomatic complexity, as a measure of code quality?

RQ1: How does prior programming experience affect cyclomatic complexity?
RQ2: What is the relationship between prior programming experience gained within different contexts (academic vs industry) and the cyclomatic complexity score?

3. Methodology

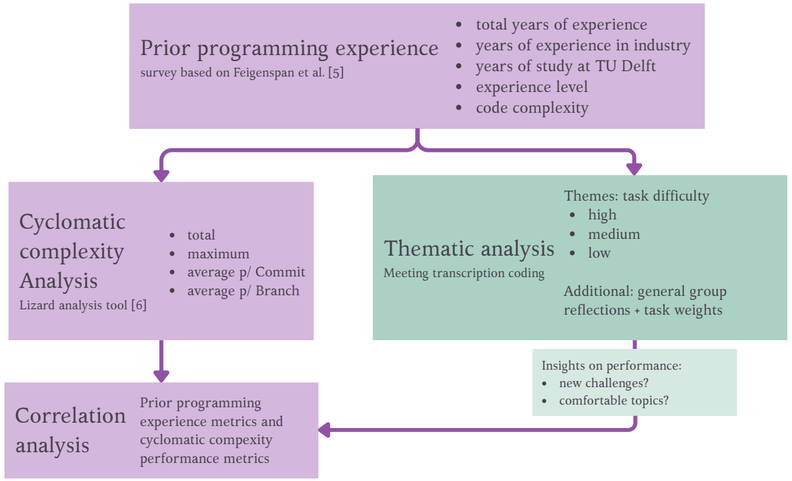

Figure 1: Methodology procedure

4. Results

RQ1: There is an expectation that prior programming experience negatively correlates to maximum cyclomatic complexity and positively correlates to total cyclomatic complexity.

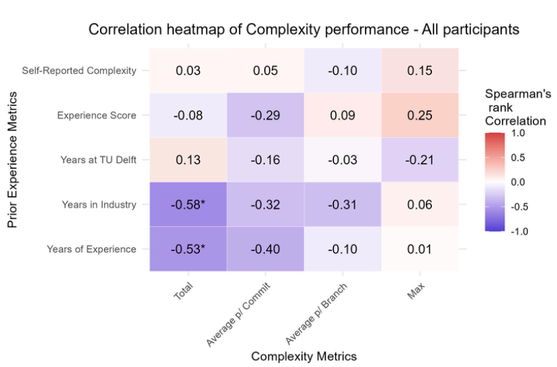

Figure 2: Correlation heatmap, all metrics vs all participants

Correlations by group

- Group A: strong negative correlations in Year based metrics. Years of Experience significantly correlated to Average p/ Commit ($r \approx -0.97$ $|p| < 0.05$).
- Group B and C: mixed results overall

Thematic analysis

Within expectations:
- Moderate/High experience → higher performance
- Low experience → lower performance

Unexpected:
- Low experience → higher performance
- High experience → lower performance

RQ2: Prior research found that experience gained in academia has a positive effect with code quality, whereas industry experience had no relationship [3]. Do these links hold when considering cyclomatic complexity, in collaborative student projects?

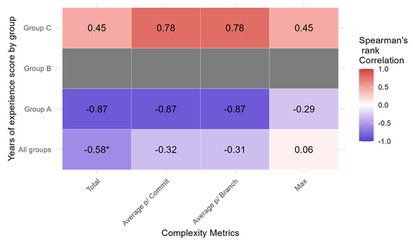

Figure 3: Correlation heatmap - Industry experience vs complexity

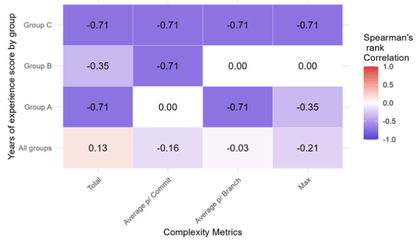

Figure 4: Correlation heatmap - academic experience vs complexity

Industry experience

- Experience in industry showed a statistically significant correlation with Total Complexity
- Different patterns of performance within different groups

Academic experience

- No statistically significant correlations found
- Patterns found within different groups disappear when considering full dataset - Simpson's paradox
- Group C: Consistent negatively correlation trend

5. Discussion & Limitations

- Students with **higher experience** tended to have **lower complexity** sums, which contradicted expectations
- Trends in task difficulty do not explain different complexity performances
- No support for experience in academia leading to better performance, when considering cyclomatic complexity
- Overall findings suggest **nuanced relationship** potentially affected by project settings (e.g., programming paradigm, size, scope) and task type (e.g., aim, front vs back-end).

- Total Complexity without task type and number limits the interpretations.
- Short time frame and late data collection
- Recruitment bias - e.g., students with higher confidence may be over represented
- Survey design - experience at TU Delft as a proxy of academic experience

6. Responsible Research

- Before the start, participant consent was collected via a consent form.
- Research has been conducted in accordance with Dutch Code of Conduct for Research Integrity
- Data will be stored within TU Delft servers and its access restricted to the Research Project team.
- The collected data will be anonymized.
- To ensure replicability, methodology was described in detail and scripts were published.
- Reproducibility not feasible due to privacy constraints.

7. Future Work & Takeaways

- Analysis of cyclomatic complexity across different programming paradigms and task types
- Testing behavioral patterns unveiled by thematic analysis in different conditions

- Interpretation of cyclomatic complexity may vary for different projects and student experience levels. Team and project contexts are important to consider when evaluating it.
- Cyclomatic complexity could have higher utility as a formative feedback tool rather than a direct assessment metric

8. References

[1] Kevin Buffardi. "Assessing Individual Contributions to Software Engineering Projects with Git Logs and User Stories". In: Proceedings of the 51st ACM Technical Symposium on Computer Science Education. SIGCSE '20. event-place: Portland, OR, USA. New York, NY, USA: Association for Computing Machinery, 2020, pp. 650–656. isbn: 978-1-4503-6793-6. doi: 10.1145/3328778.3366948. url: https://doiorg.tudelft.idm.oclc.org/10.1145/3328778.3366948.

[2] Christof Ebert, James Cain, Giuliano Antoniol, SteveCounsell, and Phillip Laplante. "Cyclomatic Complexity". In: IEEE Software 33.6 (Nov. 2016), pp. 27–29.

[3] 'Oscar Dieste Tub'ıo, Alejandrina Aranda, Fernando Uyaguari, Burak Turhan, Ayse Tosun, Davide Fucci, Markku Oivo, and Natalia Juristo. Empirical evaluation of the effects of experience on code quality and programmer productivity: an exploratory study". In: ICSSP '18: Proceedings of the 2018 International Conference on Software and System Process. May 2018, pp. 111–112. ISBN: 978-1-4503-6459-1. DOI: 10.1145/3202710.3203163.

[4] Jefferson Lopes, Johnatan Oliveira, and Eduardo Figueiredo. "Evaluating the Impact of Developer Experience on Code Quality: A Systematic Literature Review". In: May 2024, pp. 166–180. DOI: 10 . 5753 /cibse.2024.28446.

[5] Janet Feigenspan et al. "Measuring programming experience" In: 2012 20th IEEE International Conference on Program Comprehension (ICPC). 2012, pp. 73–82. doi: 10.1109/ICPC.2012.6240511.

[6] Terry Yin. terryyin/lizard. original-date: 2012-06-21T11:31:46Z. May 2025. url: https://github.com/terryyin/lizard (visited on 05/17/2025).