Correlation Between Prior Programming Experience and Task Distribution in Student Software Development Projects

Jocelyn Woods [1] Supervisors: Fenia Aivaloglou [1] and Merel Steenbergen [1] Examiner: [1]

[1]EEMCS, Delft University of Technology

Background and Motivation

Group based software development courses are at the core of computer science education curricula.
Not all students **contribute equally** to these projects. This phenomenon is often referred to as **social loafing** [1].
One potentially important influence, the **correlation** between a students **prior programming experience** and **contributions**, has been under researched.
If prior experience is relevant to task distribution, TAs, lecturers, and teammates could account for this during planning so that each member can benefit.
The correlation between prior programming experience and task distribution will be analyzed **quantitatively** and **qualitatively**.

Research Questions

What is the effect of prior programming experience on the task distribution in student software development projects?

How does **prior experience** correlate with the **types of tasks** team members are assigned?

What relationship does more **prior experience** have on students likelihood of taking on more **difficult or time consuming** tasks?

What is the difference between **tasks assigned** to each team member and **tasks done** by each team member?

Methodology

Survey – Students have filled in a survey created by extracting questions from a tested survey regarding prior experience [**?**]

Thematic Analysis – Transcripts from multiple sprint meetings of three different software-engineering groups were coded to analyze conversations about task distribution.

Quantitative Analysis – Data about issue distribution and difficulty will be gathered and compared to the survey results to examine possible correlations.

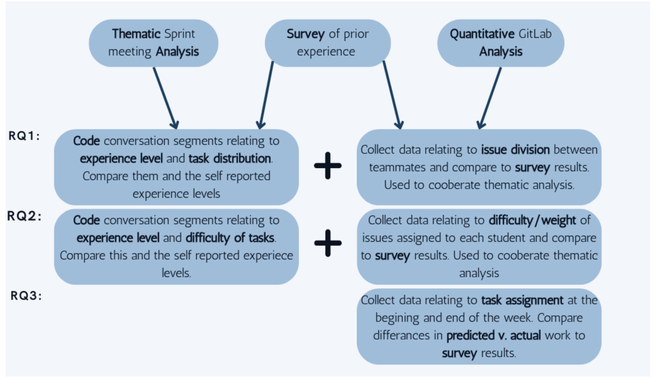

Figure 1. Observation Methodology

Results

Quantitative Analysis Results

Metric	Years Study	Years Programming	Experience Self	Experience Relative
AssignCreate	-0.16 / 0.493	-0.21 / 0.378	-0.01 / 0.957	-0.10 / 0.690
Documentation	0.08 / 0.737	0.35 / 0.125	0.12 / 0.604	0.09 / 0.694
FrontBack	0.20 / 0.402	-0.22 / 0.345	-0.14 / 0.555	-0.12 / 0.628
Difficulty	-0.12 / 0.622	-0.16 / 0.504	0.04 / 0.855	0.09 / 0.721
Reassignments	-0.10 / 0.651	-0.04 / 0.873	-0.17 / 0.453	-0.12 / 0.594
GiveAway	-0.24 / 0.274	-0.14 / 0.537	-0.30 / 0.172	-0.26 / 0.246

Table 1. Pearson correlations (r / p) between experience predictors and task-related metrics. Color intensity reflects strength and direction of correlation: blue = positive, red = negative.

Metric	Years Study	Years Programming	Experience Self	Experience Relative
Managerial	0.07 / 0.767	-0.20 / 0.391	0.15 / 0.524	-0.05 / 0.832
Assign/Create	0.14 / 0.550	0.22 / 0.362	0.11 / 0.649	0.06 / 0.786
Technical	0.39 / 0.089	-0.22 / 0.358	-0.14 / 0.562	-0.11 / 0.631
Difficulty	-0.09 / 0.710	-0.04 / 0.856	0.08 / 0.731	0.05 / 0.835
Gave Away Count	-0.17 / 0.443	-0.11 / 0.631	-0.33 / 0.134	-0.25 / 0.269
Reassign Count	-0.08 / 0.739	0.03 / 0.896	-0.13 / 0.578	0.01 / 0.979

Table 2. Spearman correlations (r / p) between experience predictors and task-related metrics. Colors match the Pearson table for comparison.

Thematic Analysis Results
For RQ1 tags related to any time a student organized the meeting or directed the programming process. A frequency analysis was preformed to find the most common contributors.

The distribution was not equal meaning there was often some students who participated very actively while others did not.

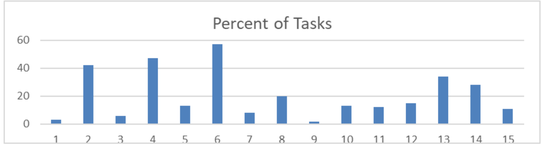

Figure 2. Distribution of managerially coded conversations by student

With first and second most common contributors often the second would speak up to contradict or point out problems missed by the first

For RQ2 If students took on a task, assigned a task, gave away a task they were working on their reasoning was considered.

While prior experience is cited as a reason for task distribution it is not the only one especially as the project progresses.

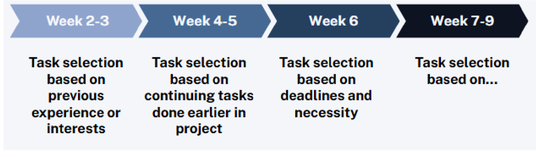

Figure 3. changes in distribution reasoning over time

Responsible Research

This research has been based on the five principles of the Netherlands Code of Conduct for Research Integrity [3]:

Honesty - All results of the study will be accurately reported and all contributing parties will be acknowledged

Scrupulousness - All methods used to gather and analyzed in this study will be supported by relevant related literature

Transparency - All data and data sources used will be clearly stated, and all results and methodology used to achieve these results will be reported and motivated

Independence - Reported results and methodology will be motivated to avoid influence by non-scientific considerations and biases

Responsibility - The privacy safety of all subjects will be considered. All data will be anonymized before reporting

Discussion

Discussion:

The quantitative analysis showed no statistically significant results which indicates a need for a larger sample size or indicates that prior experience is not an influencing factor in task distribution
The thematic analysis dis find some relation between experience and task distribution but also indicated additional factors like interests and time pressure

Possible limitations:

Limited groups were recruited to participate in this study
Inability to follow the software project to completion due to time constraints
Biases due to volunteer based recruitment

Future work

Further research into these topics could likely benefit students and teaching staff alike. Some possible recommendations would be

A larger scale study will a larger number of participants from multiple universities following the students throughout the entire process.
Conducting a multivariate analysis to further investigate the correlation with prior experience metrics.
Investigating alternative causes for social loafing such as gender, personal disposition, and other group dynamics.

References

[1] Ilenia Fronza and Xiaofeng Wang. Towards an approach to prevent social loafing in software development teams. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 241–246, 2017.

[2] Laura Heels and Marie Devlin. Investigating the role choice of female students in a software engineering team project. In *Proceedings of the 3rd Conference on Computing Education Practice*, CEP '19 2, Durham, United Kingdom, 2019. Association for Computing Machinery.

[3] Netherlands Code of Conduct for Research Integrity. Netherlands code of conduct for research integrity. `https://www.nwo.nl/en/netherlands-code-conduct-research-integrity`, n.d. Accessed: 2025-06-02.