A Software Engineering Capstone Infrastructure that Encourages Spreading Work Over Time and Team

Spencer Smith, Christopher Schankula, Lucas Dutton and Christopher Anand

Computing and Software Department, McMaster University

smiths@mcmaster.ca

How can instructors spread out the work in computing capstone courses across time and among team members? We propose using a GitHub template that contains all the initial infrastructure, including the folder structure, text-based template documents and template issues. In addition, we propose each team begins the year by identifying specific quantifiable individual productivity metrics for monitoring, such as the count of meetings attended, issues closed and number of commits. To measure the effectiveness of our intervention on the distribution of work, we introduce a new fairness metric.

1. Proposed Process and Infrastructure

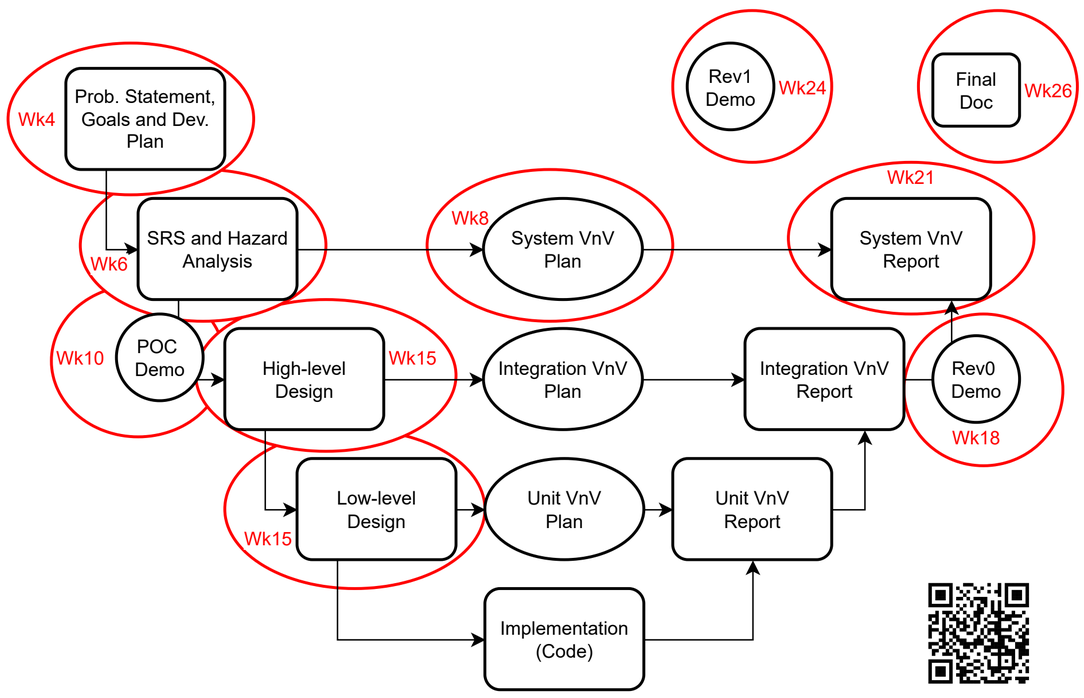

Figure 1: *V Model Used for Capstone Deliverables*

In an 8-month capstone course, teams of 4–5 follow a V-model process. They propose a team charter that includes specific quantifiable expectations and consequences. Team member productivity is reported to the instructor. A template Github repository is used for standardization and to save time.

2. Time Spread Before and After Templates

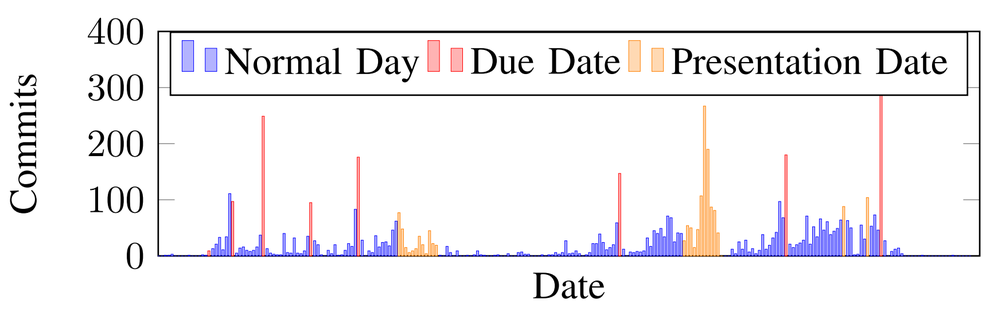

Figure 2: *Commits for 2022–2023.*

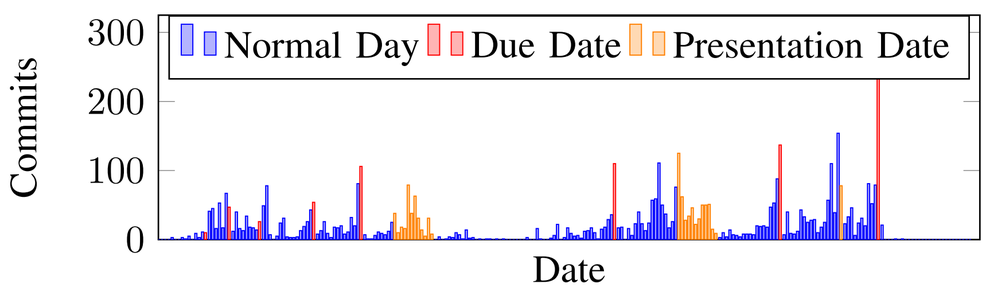

Figure 3: *Commits for 2023–2024.*

The data shows the benefit of partial implementation of the proposed intervention, but there is still room for improvement. Full elimination of the deadline effect is unlikely because of the nature of due dates in students' busy schedules.

Table 1: *T0 on deadline, T2 up to two days prior*

Metric	2022/23 Value	2023/24 Value
Total Commits	6140	5120
T0 Commits	1471 (23.96%)	942 (18.40%)
T2 Commits	2377 (38.71%)	1872 (36.56%)

Team Fairness Before and After Templates

$$\text{fairness}(C) = 1 - \frac{\sum_{c,x \in C, c>x} (c - x)}{(|C| - 1) \cdot \sum_{c \in C} c}$$

where C is the multiset of teammates' numbers of commits to the repository. The index computes one minus the sum of the difference between each teammate's commits and those who committed less than them, normalized by the number of teammates (excluding themselves) and the total number of commits. This yields a value from 1 to 0, called the *fairness* index.

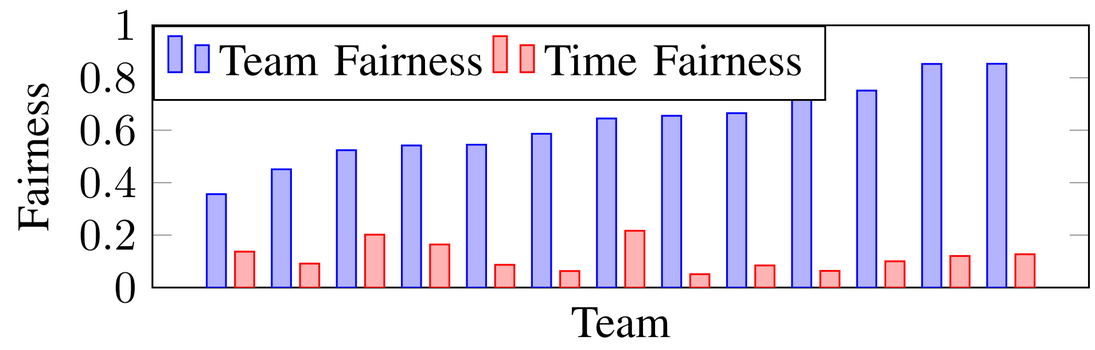

Figure 4: *Fairness of Commits 2022/23 [n=13; Team Fairness Mean: 0.63, Stddev: 0.15; Time Fairness Mean: 0.12, Stddev: 0.05; Correlation: -0.16]*

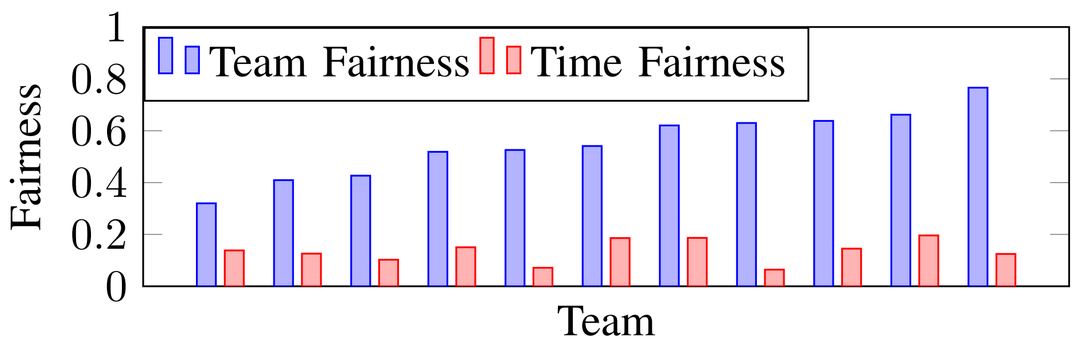

Figure 5: *Fairness of Commits 2023/24 [n=11; Team Fairness Mean: 0.55, Stddev: 0.13; Time Fairness Mean: 0.14, Stddev: 0.04; Correlation: 0.15]*

In the future

- Experiment with applying new index to other metrics like lines of code, issues created/closed, pull requests created/merged, etc.
- Use multiple metrics at once, similar to the multi-Jain fairness index
- Investigate correlation between lower fairness values and perceived unfairness by the teammates
- Investigate whether live access fairness index encourages teams to share work or simply encourages them to "game the system"

3. Conclusions

Preliminary data suggests that templates, team charters and productivity monitoring may have an impact. In the base year we observed 24% of commits happening on the due dates, but after partially introducing the proposed interventions this number improved to 18%. Going forward, we propose an experiment where commit data and interview data is compared between teams that use the proposed interventions and those that do not.

CSEE&T 2025, IEEE Conference on Software Engineering Education and Training, Apr 28–29, Ottawa, ON, Canada