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

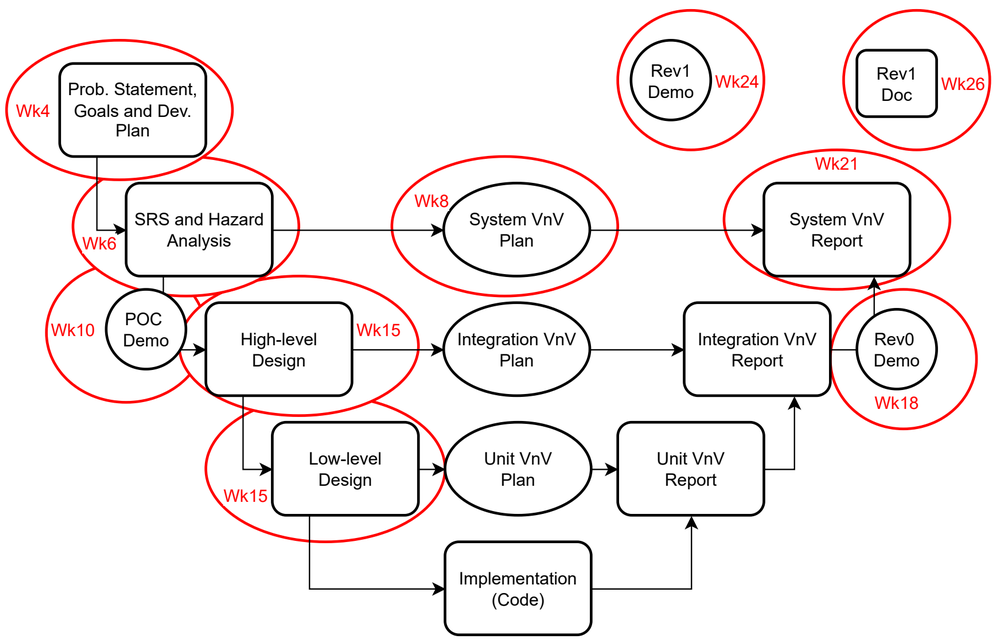
Spencer Smith, Christopher Schankula, Lucas Dutton and Christopher Anand

Computing and Software Department, McMaster University

smiths@mcmaster.ca

How can instructors facilitate spreading out the work in a software engineering or computer science capstone course across time and among team members? we propose using a GitHub template that contains all the initial infrastructure a team needs, 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.

1. Proposed Infrastructure



Github template (link), quantified team charter.

2. Time Spread Before and After

Time-Spread Metrics Across Two Classes (T0 on deadline, T2 up to do 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%)

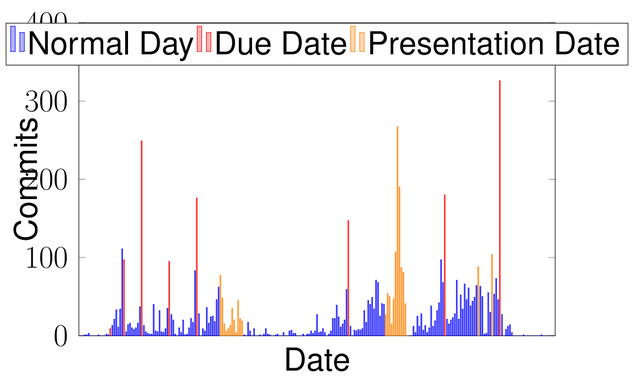


Figure 2: *Histogram of Commits for 2022–2023. Dates shown in red are due dates for major written deliverables, and dates in orange are days where presentations were scheduled.*

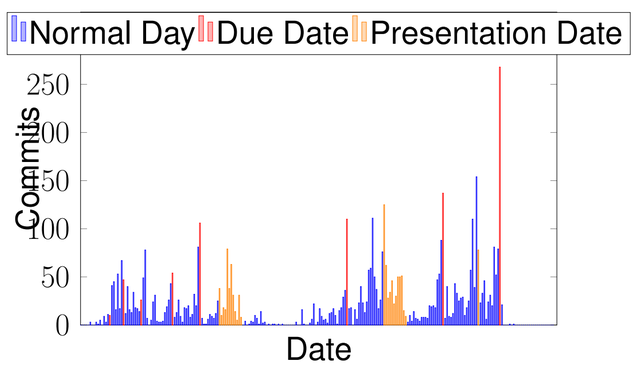


Figure 3: *Histogram of Commits for 2023–2024. Dates shown in red are due dates for major written deliverables, and dates in orange are days where presentations were scheduled.*

Team Fairness Before and After

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

where C is the multiset of teammates' numbers of commits to the repository. The index computes 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 0 to 1, called the *unfairness* index, where:

- 0 indicates that teammates did an equal amount of work
- 1 indicates that all the work was done by one teammate
- A value between 0 and 1 indicates the proportion of work per person which could have been given to someone who did less work

Fairness is defined as $\text{fairness}(C) = 1 - \text{unfairness}(C)$.

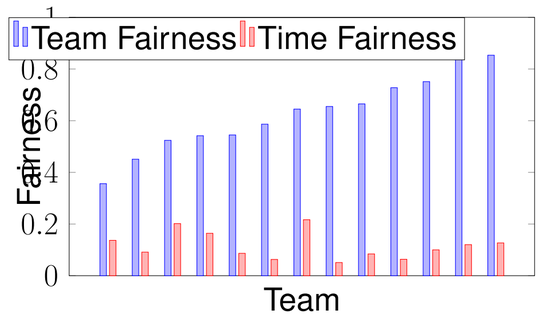


Figure 4: *Fairness of Commits Per Team 2022/23 [n=13; Team Fairness Mean: 0.63, Stddev: 0.15; Time Fairness Mean: 0.12, Stddev: 0.05; Correlation: -0.16]*

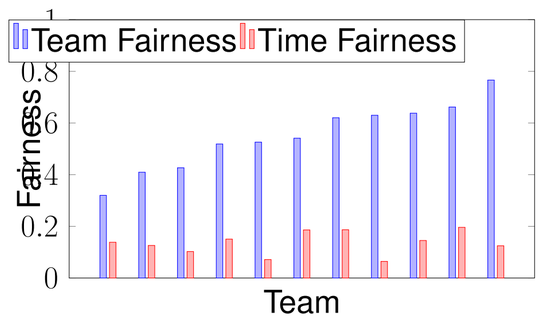


Figure 5: *Fairness of Commits Per Team 2023/24 [n=11; Team Fairness Mean: 0.55, Stddev: 0.13; Time Fairness Mean: 0.14, Stddev: 0.04; Correlation: 0.15]*

3. Conclusions

Preliminary data suggests that these steps may have an impact. In 2022 – 2023 we observed 24% of commits happening on the due dates. After partially introducing the above ideas in 2023/24, this number improved to 18%. To measure the fairness we introduce a fairness measure based on the disparity between number of commits between all pairs of teammates. 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