University of Toronto
Department of Computer Science

CSC302H – Engineering Large Software Systems

Instructor: Matt Medland
March 16, 2014

Assignment 4: Build & Test a New Feature

Due Date: Wednesday, April 2, 2014 – 11:59 p.m.

This assignment counts for 10% of the final grade.

Build and test the *matplotlib* feature for which you performed a requirements analysis in A3. You will be tracking your progress using an Agile Horizon Plan (a.k.a. "release" or "development" plan) as discussed in section 3.4 of the Penny book. You will break your chosen feature down into implementable chunks, and you will track progress on your plan, keeping capacity & requirement in balance as you go. In addition to the plan, you will provide a burn-down chart showing progress over the course of the assignment. You are required to write test cases and demonstrate the code coverage of those tests. During the last lecture/tutorial, each team will conduct a 5-minute demo of their implementation. Each team will submit a single report containing the items mentioned above.

Doing the Assignment:

This assignment has 8 steps. They are:

1. Bring your group's Git repo on CDF up to date with the *matplotlib* project's master branch in *github*. This should be done as early as possible, and certainly before any code is committed for your chosen feature. Document in your report how this process went including any auto-merge failures and how you resolved them.

2. Split up your chosen feature into implementable chunks and play planning poker with your group to estimate the effort required. When estimating use real time units, like hours or days, rather than something abstract like points. Are the estimates different from what you thought when estimating in A3? If so, why? Discuss this in your report. Use a tool to track your list of broken-down feature items. Microsoft Excel is acceptable for this, but feel free to choose other tools that are designed specifically for issue tracking.

3. Decide on your development process and how you will organize and track your implementation efforts. Decide whether any of the software process models discussed in the course will help you. For example, consider whether to use the following strategies: TDD, pairwise programming, daily scrum meetings, waterfall, etc.

4. Make a release plan as outlined in section 3.4 of the Penny book. Keep the plan up to date as progress is made. Your estimates from step 2 will make up the requirement for the plan (F) and developer time available will make up your

team's capacity (N×T). Your release date is the due date of this assignment. Along with the plan, you will maintain a burn-down graph showing the historical progress of the plan. Whenever you update estimates or log work completed you should update the plan and the burn-down graph.

5. Create unit (white-box) tests, and automate running the tests, for your new feature. Depending on your development strategy, you may write your test cases before your features, as you go, or after your features are complete. Discuss the level of code coverage of your test suite in your report.

6. Write a report documenting your re-synch with the *github* project, feature break-down and sizing exercise, development process, release plan, burn-down graph, and unit tests. Describe what went well, and what did not go well, and anything unexpected that occurred. For example, were there any feature estimates that were off by a significant amount? Did you have to adjust the plan by dropping any feature items, etc.

7. Prepare a 5-minute feature demo that will be presented during the last class. Prepare one or two slides and a brief live demo will be sufficient.

8. Document your teamwork by completing the group evaluation form available on the course webpage. Submit your printed peer evaluation form in person to your TA at the beginning of the interview.

What to Submit:

Submit your report electronically on CDF, or by email to the instructor only if there is a problem submitting on CDF. The report should not exceed twenty (20) pages (not counting cover pages, appendices, and group evaluation forms). It should include the following items:

1. A discussion of the re-synch with the *matplotlib* master branch in *github* as outlined in item 1 under "*Doing The Assignment*" above.

2. A discussion of your feature breakdown and estimation activities (using "planning poker") and if/why the estimates were different from those reported in A3. A release plan, containing the features & estimates, as it looked at the beginning of the development effort, and the resulting plan at the end of the effort. Include the final burn-down graph, showing important events on the graph like capacity constraint violations, major feature resizing (up or down), and rebalancing events like cutting features.

3. Your chosen development methodology/process and its (in)effectiveness.

4. A discussion of your automated unit tests, and their associated code coverage.

5. A general "post-mortem" discussion of what went well, what unexpected things came up during the development effort, and any adjustments you had to make to the plan as a result. Also, what you would have done different if you had it to do over again.

Be sure to include a cover page indicating the name of your team, the names and student numbers of all team members, title of work, course, and date. Assignments will be judged on the basis of visual appearance, the grammatical correctness and quality of writing, and the visual appearance and readability of the plan and graphs. Please make sure that the text of your report is well-structured, using paragraphs, full sentences, and other features of a well-written presentation. Use itemized lists of points where appropriate. Text font size should be either 10 or 12 point.

Marking Scheme:

Your assignment will be graded by your TAs. If you have questions about a marked assignment, you should first ask your TA before/after a tutorial or by email. If you don't get satisfactory answers, you should talk to your instructor.

Marks for this assignment will depend on the following factors:

- **Re-synching your project with the master branch in *github* (5%):** Did any issues come up that caused you trouble? Did you describe them well and how you handled them? Were their problems merging? Did you have any issues with missing python libraries? If the process went perfectly smooth, and the auto-merge was flawless, there is no need to elaborate.

- **Feature break-down & development plan tracking (30%):** Describe the process of breaking your feature down into implementable chunks of work. Did you describe your planning poker results and differences (if any) from A3 estimates? Did you construct an initial version of the development plan that was balanced (i.e. $F \leq N \times T$)? Did you use an appropriate tool to track your features? Did you adjust your estimates as work progressed and you learned new information? Did you re-balance your release plan if necessary and justify your changes? Did you keep a burn-down graph updated as you made changes to the release plan? How did the burn-down graph help you make decisions regarding how to adjust the plan?

- **Discussion of your development process (15%):** Did you describe your chosen development process? Was it effective? Why/why not? What changes (if any) would you make if you had to do it again? What were the states that you chose to track your features (ex. new, assigned, finished, accepted) and why?

- **Automated testing (20%):** Did you write an appropriate set of unit tests? Did you automate the execution of the tests effectively? Did you provide an analysis of the code coverage of the unit test suite?

- **Professional quality report (10%):** The style of your presentation, including language, grammar, clarity of the presentation, layout and legibility of the diagrams, charts, graphs etc. (5% - Language; 5% - Style and clarity).

- **Demo Presentation (10%):** Was your demo presentation clear and to the point? Was it obvious to the audience what was the new functionality? Was it clear why this new functionality is valuable to users?

- **Interview (10%):** Did you work eﬀectively as a team? Did you demonstrate active involvement of all team members in the project? If your teams had problems, did you address them eﬀectively and in a timely manner?