University of Toronto
Department of Computer Science

CSC302S – Engineering Large Software Systems

October 10, 2008
Prof. Steve Easterbrook

Assignment 2: Implementing Change Requests

Due Date: 1:20pm, Monday, October 27, 2008
(i.e. within 10 minutes of the start of the lecture)

This assignment counts for 10% of the final grade

Analyze a list of change requests for the JEdit application. Using an appropriate software development process, select at least two change requests from the list, complete the implementation of them, and provide test cases suitable to demonstrate that the changes have been correctly implemented. The assignment requires you to use your judgment about what software process to use, and which change requests to select for implementation. There is no "correct" choice – you will be given credit for selecting change requests that can be implemented correctly in the time available, and which are most likely to satisfy the users. Note that users are more likely to value simple fixes that work reliably over more ambitions features that are incomplete. For this assignment, assume that your TA and instructor are representative of your target users

The project is to be carried out in your assigned teams. Each team will submit one report.

I. Doing the Assignment

This assignment has 8 steps. They are:

1. *Read the prioritized list of change requests provided on the next page*. Check that you understand what each change request is asking for. Ask your TA or the instructor if you need any user clarifications.
2. *Draw a Use Case Diagram illustrating all the Use Cases relevant to this list of changes*. Use an appropriate UML drawing tool to draw this diagram.
3. *Select an appropriate software process model to guide you through the remainder of this assignment*. Use any of the software process models presented in class (e.g. SCRUM, XP, ICONIX, RUP), or another that you are familiar with. You will need to consider how to adapt the process to your particular needs.
4. *Plan a new (minor) release of JEdit by selecting an appropriate subset of the list of change requests on the next page*. Use your chosen process model to guide you. This may involve documenting the Use Cases in more detail. It may involve estimating the effort required to implement each change, and identifying any anticipated risks. It should involve some method for determining which team member is allocated to work on which task. Note: your planned release must include implementation of ***at least two*** of the requested features.
5. *Implement your selected change requests*. Be sure the check the edited code back into your code repository when it is ready. Your TA will check out the code from your repository to run it, when marking this assignment. Make sure you ***clearly*** indicate what to check out!

6. *Write test cases to demonstrate that the changes have been implemented correctly*. Design these as "customer acceptance" tests – i.e. a description of the steps a user needs to carry out to check that the feature works as requested.

7. *Write a report* that describes the steps you went through to select and implement the change requests that were included in your new version of JEdit. Be sure to document your development process, and comment on how well the process worked for you.

8. *Document your teamwork* by completing the peer review forms on the course website.

II. List of change requests

The following list of nine change requests has been sorted into (approximate) priority order, with the highest priority (most requested) changes listed first:

1. The pulldown menus indicate that all commands are available, even when some are not. Menu commands that are not currently available (e.g. folding commands when folding is turned off) should be grayed out, to indicate they cannot be selected.
2. JEdit has no command to list all open views, and hence it can be hard to switch quickly to a different view when a large number of views are open. The easiest way to achieve this would be to add a menu item that lists all current views as a submenu, allowing any to be selected.
3. When turning Folding on as a global option, the change does not affect current buffers, only new ones created or opened after the change. Changing the global option should affect all current buffers too.
4. In the "Save as" dialog, when saving a new file for the first time, the dialog suggests a name for the file. It would be useful if it also automatically appends an appropriate file extension (using the current edit mode to determine file type).
5. If you close and then reopen the "Find and Replace" dialog box, the string last used for "find" is still there, but the string used for "replace" is gone. They should both reappear.
6. The command "Select fold" behaves still works when folding is turned off, but will only select the highest-level fold. When folding is turned on, it selects the lowest-level fold. It should behave consistently.
7. JEdit has a feature to dock some windows within a view. It then provides commands to undock and to close the docked window. However, undocking currently also closes the docked window. Undocking should not close the docked window.
8. If you open several views, the mouse pointer no longer changes to an arrow when you move it over the buffer switcher for all views (the most recently opened view works correctly, the others don't). The cursor should change to an arrow for all views.
9. For the command "Close all buffers", it would be useful for JEdit to ask for confirmation, because if you accidentally close a large number of open buffers, it's a pain to re-open them all.

II. What to Hand In

Hand in your report at the start of your lecture on the due date. *Reports not handed in within the first ten minutes of the lecture will be treated as late*.

The report should not exceed twenty (20) pages (not counting cover pages, and appendices). It should include the following items:

1. A brief description of the software development ***process*** you used, including the reasons you selected this process, and any steps you took to adapt the process to your needs.

2. A Use Case diagram, plus any other documentation you produced to describe Use Cases and/or change requests in more detail.

3. A brief description of the plan you created for your new version of JEdit. Describe the rationale you used for selecting the changes you chose to include and any risks you identified when you developed the plan. Write a brief technical commentary on how the changes affect the design and/or the code of JEdit,

4. A set of customer acceptance tests, described in a form that would allow any user to download the new version of JEdit, run the application, execute the tests, and determine that the software works correctly.
5. A review of lessons learnt in carrying out this assignment, including commentary on how the chosen process helped or hindered you, and any problems you encountered.

Written Presentation Requirements

Be sure to include a cover page indicating the name of your team, the names of all team members, title of work, course, date and tutor's name. 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 models, as well as their contents. 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.

IV. Marking Scheme

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

Marks for this assignment will depend on the following factors:

Description of your process (20%): Did you identify and evaluate a suitable development process? Does your choice take into account the circumstances of this project, including project size, team experience and schedule? Did you clearly describe how you adapted the process to your specific team's needs? Did you understand how to apply the process, and did you follow it? Did you describe how well the process worked, and identify lessons learnt?

Description of Use Case Analysis (20%): Did you identify an appropriate set of use cases for the given list of change requests? Did you draw a Use Case diagram? Are your use cases written from the users' perspective? Did you provide additional descriptions of the use cases as appropriate to your chosen process model?

Your plan for the new version of JEdit (20%): Did you select a manageable subset of the change requests? Did your plan take into account the user's prioritization, as well as the time and effort available? Did you clearly state the rational you used for this selection? Did you identify the major risks associated with your plan? Did you follow the plan, making any adjustments to the plan as needed? Did you describe what the changes were, and how they affected the code?

Working application and test cases (20%): Is it clear which version in the repository constitutes the new release? Can the new version of your software be checked out from your repository and does it run? Are your test case clearly described? Can a user execute the test cases without any of your developers being present? Do the changes work the way they should?

Presentation (20%): The style of your presentation, including language, grammar, clarity of the presentation, layout and legibility of the diagrams, etc. (10% - Language; 10% - Style and clarity)