University of Toronto
Department of Computer Science

CSC302S – Engineering Large Software Systems

January 31, 2012
Prof. Steve Easterbrook

Assignment 2: Implementing Change Requests

Due Date: 10:20am, Thursday, February 23, 2012
(Note: this is during reading week.
Submit to Prof Easterbrook in BA3259)

This assignment counts for 10% of the final grade

Analyze the change requests and bug reports for one of the OpenStack components. Using an appropriate software development process, select at least two items (*bug*s and/or *blueprints*), complete the implementation of them, and provide test cases suitable to demonstrate that the changes have been correctly implemented. *Note: For this assignment, you may select items that are already listed as implemented in Launchpad. In that case you will still be expected to complete all the analysis, and to devise your own test cases to check the implementation works as expected.*

The assignment requires you to use your judgment about what software process to use, and which change request to select for implementation. There is no "correct" choice – you will be given credit for selecting changes 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.

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. *Explore the lists of bug reports and blueprints in the project Launchpad for an OpenStack component of your choice*. Be sure to explore the lists for both the Diablo and Essex releases. Select a handful of interesting bug reports and/or feature requests to explore further.
2. *Draw one or more Use Case Diagrams illustrating all the Use Cases relevant to the items on your shortlist*. Use an appropriate UML drawing tool to draw these diagrams.
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. *Select at least two items from your shortlist to implement and test*. Use your chosen process model to guide you in the selection process. 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 members are allocated to work on which task.
5. *Implement your selected items*. Be sure the check the edited code into your code repository on DrProject 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 items as requested.
7. *Write a report* that describes the steps you went through to select and implement the items you worked on in this assignment. 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. What to Hand In

Hand in your report to the instructor by 10:20am on Thursday Feb 23rd. *Reports not handed in by this time 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 showing use cases relevant to the list of bugs/blueprints on your shortlist, plus any other documentation you produced to describe Use Cases and/or change requests in more detail.
3. A brief description of your implementation plan. Describe the rationale you used for selecting the items you chose to work on, 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 OpenStack,
4. A set of customer acceptance tests, described in a form that would allow any user to download your new version of an OpenStack component, run it, 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 tutor will mark your assignment. 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 draw up a shortlist of candidate blueprints/bugs? Did you identify an appropriate set of use cases for the items on your shortlist? 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 implementation plan (20%): Did you select a manageable subset (but at least two) of the items on your shortlist for implementation? Did your plan take into account user's likely priorities, 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)