Group Project

Re-engineering an Open-Source Software System

Version 1.1.0 (10/5/2022)
Changes are highlighted in red.

For this assignment you must, as a group, re-engineer some aspect of the system that you have been studying as a group.

This assessment counts for 20% of your overall module mark.

Setup

Note on groups

For the project, you must have a minimum of three group members. Please make sure that you contact your group members as soon as possible. You can use the Blackboard group email facility to do this.

In the event that there are unresponsive team members, or where it is known that team members have left the course, please contact Richard Somers (rsomers1@sheffield.ac.uk). He will establish whether the member has indeed left the course, and reallocate remaining members to other groups (or bring in members from other groups) if necessary.

The assessment will vary when we consider the individual marks for the presentation. If an individual covers several topics we will give each an independent mark, and pick the highest one to feed into the final overall mark.

Setting up GitLab

You should all have access to your groups via GitLab. You all have “Developer” permissions, which means that you are able to create new projects within your group.

Create a new, blank repository called “SubjectSystem”, with private visibility to your group.

Clone your actual group software system that you’ve been working on from GitHub into a fresh directory.

In the cloned repository, use git to remove the references to its host repository.

```
git remote rm origin
```

Add your newly created GitLab repository as a remote repo:

```
git remote add origin https://stugitlab.dcs.shef.ac.uk/courses/com3523/2021-22/test-group/test.git
```

The above command refers to a test repo for the purposes of illustration - replace the git address with your newly created git repo.

Push the contents of your cloned repo up to your newly created repository:

```
git push -u origin —all
```

You can now use this repository as a basis for your re-engineering.

Important: Do not pull over any subsequent remote changes to the repository. When we look through the commits to this repository, the most recent ones must be the ones that you have done to re-engineer it.

Tasks

(1) Identify a weakness with the structure of the source code in the system.

Drawing on the evidence that you have collected so far, using the various analysis techniques that we have explored, identify a weakness within the design of the source code.

It is probable that you will identify several candidate problems with the source code in the system. In order to pick the weakness to focus on for this assignment, it should be non-trivial in nature, requiring several re-engineering steps.

As a guide, simply splitting up a method into smaller methods using the “Extract Method” refactoring would be too trivial. However, moving the method into its own Object (as discussed in Week 10), or moving behaviour closer to data to eliminate navigation code, etc., would be appropriate.

(2) Write regression tests.

Do not use existing test cases. Generate 2-3 test cases that execute code that will be re-engineered. If you believe that Mock testing will help, use Mockito. Commit your test cases to the version repository.

(3) Plan and attempt the re-engineering process. Do so via a series of commits to the version repository. Describe any problems you encountered.

For some objectives, completely eliminating the design weakness may require more time and effort than is feasible for this assignment. Your submission will be assessed in terms of (a) the techniques that you have successfully demonstrated, (b) their appropriateness. So you can choose an ambitious reengineering task, but do not have to follow it through to completion, just as long as you demonstrate the key steps involved.

For all commits that are related to the re-engineering of the source code (and setting up the test set) , add “REENG: “ to the beginning of the commit message.

(4) Reflect on the impact of the reengineering effort. If possible, gather evidence from the reengineered system to support that the reengineering effort has had a positive impact.

Format and Submission

The submission will include the following components: A set of GitHub commits containing the re-engineering steps that were used to re-engineer your system. A presentation, submitted as a video, and an accompanying slide-deck. There will be no written report. As a group, you must create a presentation that covers the following topics (in this order):

- **System under analysis:** Give an idea of what the system does (or claims to do), how popular it is. Are there any superficial reasons why it might require re-engineering (i.e. reasons that do not require prior knowledge about the underlying implementation)? Is it a “legacy” system? Are there known problems? [1 slide max]

- **Weakness:** What is the problem with the system? Your justification should include at least 3 of the families of techniques that were covered in the first half of the course. Detail how you applied them. [3 slides max]

- **Regression tests:** Describe what you had to do to set up the test harness. Were any testability transformations required? What test cases did you choose, and why? [3 slides max]

- **Reengineering steps:** What strategy did you adopt to achieve your objectives? Which patterns did you apply, and why? [4 slides max]

- **Outcome:** What was the outcome of the reengineering efforts? Are you able to show any evidence that the results were worthwhile? Such as improvements in metrics? [1 slide max]

Submission

- Create a directory called `groupPresentation’ in the root of your group GitHub repository, and submit:
  - A single PDF file (slides.pdf) containing all of the slides that were used for the presentation.
  - We will search for commits in your repository with commit messages that start with “REENG” to identify what you have changed.
- Record your video on your group’s Blackboard Collaborate.
  - *Do not push the file to GitLab, to avoid overloading the GitLab server.*
  - If you need to edit the video, or wish to produce a video with a different tool, then you must upload the edited version to the Group’s File Exchange area.
    - Note - the file size limit is apparently 125MB, so you may need to use appropriate compression on your video.
- Submit your peer feedback to Buddy-Check on Blackboard. This will be made available closer to the submission deadline.

Ensure that both recording and GitHub commits are completed and pushed respectively by the deadline. In the case of GitHub give yourselves several days to avoid any last-minute panics.

Late submissions will be subject to the usual departmental late submission penalties, applied to the whole group.

Submissions will not be accepted via email.

Limits and constraints

The presentation should be a maximum of 10 slides long (excluding slides that contain no text and only include visuals to support what you are saying in this). The presentation should take anywhere between 8-12 minutes to present.

For your analysis, you must only use the tools that we have used in Bash or the Reengineering Toolkit.

You *are* allowed to add your own extensions to the Reengineering Toolkit, e.g. adding your own metrics.

Recording the presentation

The presentation should be recorded via Blackboard Collaborate, using the “Record” function. To record the presentation there are two options:

1. The preferred option is to do the recording “live”, using one group member to present the slides and flick through them, and allow other members to talk over them.

2. You can all record your own presentations (also on Blackboard Collaborate if you like) and then combine all of the videos into one long video. This will require some video editing capability (we will not accept multiple separate videos).

You are permitted to minimally edit the videos to remove long silences or clear errors (but don’t worry about production quality!).

For the presentations you do not need to activate your webcam if you would prefer not to. But we do need to hear your voice.

Assessment

Your grade will be worked out as follows:

The group will be assessed on:

- The technical achievements. The ability, as a group, to successfully overcome the various technical hurdles involved in analysing and re-engineering the system.
  [20%]

- The strategy of the reengineering effort. Did you justify your decisions when it came to properly testing and reengineering the system?
  [20%]

- The coherence of the presentation as a whole, and time-keeping.
  [20%]

Individual group members will be assessed on:

- Attendance in lab sessions throughout the course (weeks 4-9).
  [10%]

- A peer-assessment, assessing your contribution to the group project (starting from the date the assignment is released). This will be executed using the Blackboard Buddycheck system.
  [30%]

Questions

If you have questions about any of the techniques, please use the discussion forum for the group project on Blackboard. Most of the questions you want to ask should already be there from the practical sessions we ran in the early weeks. If not, please feel free to pose new questions!

If you have questions about the assignment that are not technical, please use the course discussion forum.

Deadline

Deadline is 15:00, 20th of May, 2022.

Support

To ensure fairness, we will only respond to queries on the Blackboard forum dedicated to this assignment. As usual, please read other queries before you post your own, to ensure that your query has not already been answered. To assist your colleagues when they are checking the forum, please give your query a descriptive title.

You must under no circumstances post your own solutions (or parts thereof) to the forum - so please be mindful when you are posing questions.

Unfair Means

It is important to bear in mind the departmental rules on unfair means. Activities such as plagiarism or collusion will be treated as a serious academic offence. This could lead to the award of a grade of zero for this assignment.

We will be closely scrutinising submissions to detect such practices, because it is important that this assessment is a genuine reflection of your own understanding of the module.

To avoid any potential accidental wrongdoings, it is especially important that you do not discuss your solutions with students in other groups. If you have questions about this assignment, please use the discussion forum.