University of Wollongong

School of Computing and Information Technology

CSCI426/CSCI926 Software Testing and Analysis

Autumn 2022

Group Project (45 marks)

- You need to form a group of **6 people** and register your group using a form on the Moodle site of the subject by the end of Week 3.
- Structure your group, allocate roles and responsibilities to your members and organize **weekly meetings**. Each group must have **at least one meeting** per week.
- Make sure your group **make progress on the project <u>every week</u>**

SUBMISSION

1. Verbal Progress report (**<u>2 marks</u>**): **Week 4 Workshop**.
    a. One member from each group presents the progress of their group in **<u>5 minutes</u>**
    b. The progress report presentation should contain:
        i. Introduce their group members and roles
        ii. Preliminary findings of all the tasks such as details about the projects, software applications and testing tools your group selects to study.
        iii. Work plans in the form of a Work Breakdown Structure (e.g. tasks, sub-tasks, who perform what, outcomes, etc.)
        iv. Meeting plans, e.g. how many meetings per week, date/time and location of meetings.
2. Verbal Progress Report **(<u>3 marks</u>)**: in **Week 8 Lecture + Workshop**
    a. One member from each group presents the progress of their group in 10 **<u>minutes</u>**
    b. The progress report presentation should contain:
        i. Reporting work completed and work in progress for **ALL Task 1, Task 2 and Task 3.**
        ii. Presenting plans for the remaining of the project.
3. Final deliverables (**<u>37 marks</u>**): **9:00am Monday May 23th 2022**
    a. Final report, README file, source code, test cases and all other artefacts your group produce for the project must be submitted to the Moodle site of the subject by one member of your group **(34 marks)**

b. Weekly meeting reports (submitted to Moodle **every week** from **Week 4 – Week 12 inclusively**) should cover at least the following:

- True group meeting records: agendas and meeting minutes which include at least the following: meeting date, attendance, progress reports, discussion summaries, and action plans/items. **(3 marks)**

4. Final presentation (**3 marks**): **Week 12 Lecture + Workshop**

IMPORTANT NOTES:

- **PLEASE READ THIS CAREFULLY:** All work should be completed **independently** by your group. Plagiarism may result in a FAIL grade and are subject to the University Academic Misconduct Procedures. If any part (including sentences, figures/diagrams, tables, definitions, descriptions, and so on) in your work are copied from other people's work (including both published and unpublished papers, reports, Web articles, etc.), or if any idea is from other people, such work or people must be acknowledged explicitly. If you directly "copy and paste" sentences from other people's work to your work, then you should use quotation marks to quote the copied sentences and make a citation next to the quoted text.
- **Member contribution for the whole project (with each member's signature)**
  - On the cover page of your group's report, you need to provide rating for the contribution of each team member and a **detailed explanation** of what the team member did for the project to justify the rating.
  - Everyone in the team should write a statement "***I agree with my group member contribution report and ratings***" and **insert their signature next to the statement**.
  - The individual contribution of each team member is assessed by all the other members.
  - The rating scale can be a **specific percentage number** (e.g. 40%, 60%, 80%, etc.).
  - Alternatively, it can be rated into one of the three scales: "contributed", "very little", and "almost no contribution". For a team member who has "contributed", he/she will receive 100% of the group mark; for a team member who contributed "very little", he/she will receive 50% of the team mark; for students who made "almost no contribution", he/she will receive 0 marks for the entire group project. Your tutor/lecturer may make adjustment to this marking criterion based on practical situations.

TASK 1: Software Quality Assurance Process (12 marks)

1. Study the software quality assurance process of one of the following open source software projects:

   a. Apache Hadoop

b. OpenStack

c. Eclipse

d. LibreOffice

e. *Your group can choose a different project to study but* ***must obtain prior approval from the lecturer****. The project needs to be similar to the above projects in terms of scale, maturity, activeness and having an existing quality assurance process in place.*

2. Provide a report detailing the existing quality assurance process of the selected project. This report needs to have **at least** the following:

    a. Details about the verification and validation activities *such as*: Who are the stakeholders and participants? When do verification and validation start? When are they complete? What particular techniques are applied during development? How is the readiness of the product be assessed? How is the quality of successive releases be controlled? How is the development process itself be improved? (***5 marks***)

    b. Details about the actual framework for software testing and analysis in place. Identifying the principles underlying the analysis and testing practice in the selected project. (***2 marks***)

    c. Details about specific test and analysis processes, techniques and activities conducted in the selected project. (***3 marks***)

    d. Any other information deemed relevant to the software quality process of the selected project, drawing from what have been discussed in this subject. (***2 marks***)

TASK 2: Software Testing and Analysis Techniques (12 marks)

1. Select a version of one of the following software applications as the subject for testing

    a. Calculator https://github.com/microsoft/calculator

    b. J2Html https://github.com/tipsy/j2html

    c. XChart https://github.com/knowm/XChart

    d. Exaile https://github.com/exaile/exaile

    e. Your group can choose a different software application to test but ***must obtain prior approval*** *from the lecturer.*

2. Provide a review of at least FIVE testing tools that can be used to test your selected application. At least one of the tools should support automatic test case generation (e.g. black- or white-box testing tools or random testing tools). A review should include **at least** a description, features, and strengths and limitations of the tools on testing the selected software application. (**5 marks**)

3. Generate 1,000 or more test cases for the selected software by yourself or using the automatic test case generation tools you reviewed. Do NOT reuse the existing test cases provided with the software application.

   Run these test cases in the sequential order and report any failure detected (***higher marks will be given to those who can detect failures***). If failures are discovered, trace them to the bugs reported in the bug/issue tracking system and the source files containing the bugs. Bonus marks will be given to new bugs discovered by your group. You need to provide evidence that the bugs are reported to the community by your group and they were accepted as new valid bugs. (***7 marks***)

4. Provide a report which clearly demonstrates how the tasks were performed in details and the outcomes of each of the above tasks.

TASK 3: Software bug dataset (10 marks)

1. Examine one of the following bug datasets:
   a. Bears https://github.com/bears-bugs/bears-benchmark
   b. QuixBugs https://jkoppel.github.io/QuixBugs/
2. Read the instructions at the dataset website to setup and run it.
3. Produce a description of the dataset, the projects, and defects found in each project (***1 marks***)
4. Choose 12 bugs from the dataset and identify the test cases relevant to those bugs. Identify which testing techniques were employed in those test cases to discover the bug (***4 marks***).
5. Modify the code to make those test cases pass. Record your attempt. (***3 marks***)
6. You can choose **one** of the following options (***2 marks***):
   a. ***Option 1***: Take your findings in **Task 2** and/or follow the same process that the authors of the selected dataset did, find new bugs.
   b. ***Option 2***: Find THREE automatic repair tools or research work that have been applied against the selected dataset. Write a report detailing features, approaches, strengths and weaknesses of those tools or research work.
7. Provide a report which clearly demonstrates how the tasks were performed in details and the outcomes of each of the above tasks.