University of Wollongong

School of Computing and Information Technology

CSCI426/CSCI926 Software Testing and Analysis

Autumn 2025

Group Project (40 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 2.

- 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>)**: submit to Moodle **Week 8 (submission deadline will be set on Moodle)**

   a. A short video (10 minutes maximum) presents the progress of your group

   b. The progress report presentation (using slides) 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.

      iii. Member contribution ratings so far.

3. Final deliverables (**<u>30 marks</u>**): submit to Moodle **9:00am Monday Week 12**

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 **(32 marks)**

b. Weekly meeting reports (submitted to Moodle **every week** from **Week 3 – Week 11 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. In particular, *report what has been done in the previous week, and what has been planned in the next week*. **(3 marks)**

4. Final presentation (**5 marks**): **Week 12 Workshop or Lecture**

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)**

  o 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.

  o 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**.

  o The individual contribution of each team member is assessed by all the other members.

  o The rating scale can be a **specific percentage number** (e.g. 40%, 60%, 80%, etc.).

  o 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 (8 marks)

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

    a. Apache Spark

    b. Apache Kafka

    c. *Your group can choose a different project to study but **must obtain prior approval from the subject coordinator**. 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.*

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

    d. 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? (***3 marks***)

    e. 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***)

    f. Details about specific test and analysis processes, techniques and activities conducted in the selected project. (***3 marks***) Note. The projects' websites include rich information. Make sure citations must be used properly in the report.

TASK 2: Software Testing and Analysis Techniques (10 marks)

1. Identify at least FIVE Web UI test automation tools. Provide a detailed review of at least FIVE testing tools from the list. A review should include an *in-depth* description, features, and strengths and limitations of the tools. (**3 marks**)

2. Select one of the following software projects as the subject for testing

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

    b. matplotlib-cpp: https://github.com/lava/matplotlib-cpp

    c. matplotlib: https://github.com/matplotlib/matplotlib

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

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 in the software projects.

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.

Provide a report which clearly demonstrates how the tasks were performed in details and the outcomes of each of this task. (*7 marks*)

TASK 3: Testing dataset and benchmarks (9 marks)

Examine the following testing dataset and benchmark:

- o SWT-Bench https://swtbench.com/ and https://github.com/logic-star-ai/swt-bench

Read the instructions at the above websites, and if possible setup and run it (single run if your PC have limited computational power).

1. (*1 marks*) Produce a description of SWT-Bench, what it is for, what it contains, what metrics it uses, and explain why those metrics are important and relevant.

2. *(5 marks)* Choose 5 instances from the dataset (one from each distinct GitHub repo) and for each instance, provide the following:

   - o Describe what GitHub repo the issue related to this instance is from. Clearly explain what the issue is. Provide the relevant code to explain and illustrate your answer.

   - o Describe the patch to resolve this issue and clearly explain why it would fix the issue.

   - o Identify the patch, fail-to-pass and pass-to-pass test cases relevant to this issue. Identify which testing techniques were employed in those test cases. Provide some examples to illustrate your answer.

3. (*3 marks*) Using the same method as described in SWT-Bench, add **3 new instances** from **a new GitHub repo** into the dataset. These new instances is from a GitHub repo of your choice, but this GitHub repo should **not** be one of the repositories in the dataset.

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

-----END OF SPECIFICATION-----