# **MILESTONE 6** -- SFT221 SCRUM Report and Reflection

All students are expected to attend the SCRUM meetings and to participate. Failure to do so will result in greatly reduced grades.

**GROUP**: 2

**Members Present**:

|  |  |
| --- | --- |
| 1. Joon Dong | 4. Heqing Xu |
| 2. Doris Chai | 5. |
| 3. Xiaopeng Liu | 6. |

## Milestone 6 Tasks

This is the final milestone where you will run the acceptance tests and fix any remaining bugs found. In addition, you will produce a testing report which lists all the tests conducted, the results and whether the bugs were fixed, and the final test passed. You will also review the test matrix to ensure every test has been performed and passed. You can change the colour of the test in the matrix to show it was run and passed. At the end, all tests in the matrix should have been passed.

The final test report can be tabular like this:

|  |  |  |  |
| --- | --- | --- | --- |
| Function/acceptance/requirement | Test Run | Bugs Fixed | Passed |
| Distance | TF001 | Did not handle negative coordinates | 🗹 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Deliverables due 4 days after your lab day:**

* Final testing report listing tests conducted, bugs fixed, and the final tests passed.
* Execute acceptance tests (results in Jira), and debug.
* Updated requirements traceability matrix stored in the repository.
* Completed scrum report including reflection questions answered.

**Rubric:**

|  |  |  |
| --- | --- | --- |
| **Individual** | Group participation (includes GitHub commits and Jira usage) | 80% |
| Teamwork | 20% |
| **Group** | Complete solution code running and executing successfully | 20% |
| Test execution (performed, results recorded, issues created) | 10% |
| Updated requirements traceability matrix | 5% |
| Final test report | 30% |
| Debugging (bugs fixed, documented, Jira updated) | 5% |
| Git usage (used properly with good structure) | 5% |
| Jira usage (creates issues, tracks progress) | 10% |
| Scrum report & reflections | 15% |
| **Deadline** | 20% deduction for each day you are late |  |

**SCRUM Report**

**Summary of Tasks Completed or Delayed in the last week:**

Here you can list all of the tasks completed in the last week along with any tasks which could not be completed with a reason why they could not be completed.

|  |  |  |
| --- | --- | --- |
| **Member** | **Tasks Completed** | **Tasks Delayed/Blocked** |
| **Joon** | **Reflection Q2, integration test case, integration test case implementation** |  |
| **Doris** | **Reflection Q1, integration test case, integration test case implementation, acceptance testing** |  |
| **Xiaopeng** | **Reflection Q3, integration test case, Update matrix traceability** |  |
| **Heqing** | **Reflection Q4, integration test case** |  |
| **All** |  | **calculateDeliveryRouteForPkg function not working, unable to output correct pathing** |
|  |  |  |
|  |  |  |

For every task delayed or blocked, describe the reason for the delay or block, how it impacts the project and the proposed solution or workaround**.**

|  |  |
| --- | --- |
| **Delayed or Blocked Task** | **calculateDeliveryRouteForPkg function not working, unable to output correct pathing** |
| **Reason for delay or block** | **shortestPath is giving a runtime error, perhaps being invoked wrong** |
| **Impact on Project** | **Unable to print correct output** |
| **Solution or work-around** | **Trying to debug shortestPath** |
|  |  |
| **Delayed or Blocked Task** |  |
| **Reason for delay or block** |  |
| **Impact on Project** |  |
| **Solution or work-around** |  |

**Summary of Meeting:**

A summary of the main points discusses in the meeting and the outcomes of the discussions.

|  |  |  |
| --- | --- | --- |
| Topic | Discussion Summary | Outcome |
| Output | **Program is still unable to print output and pathing** | **Need to debug shortestPath** |
| Final test report | **How we are going to distribute the final report** | **All members will be add their test suites to final report** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Summary of Decisions Made:**

This will include major architecture and design decisions, testing decisions, prioritization of tasks, dealing with problems encountered and other major outcomes from the meeting.

|  |  |
| --- | --- |
| Decision | Rationale |
| The shortestPath should be tested | If we can implement a test suite of shortestPath, we can get a better understanding of how we should invoke it and find potential bugs |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Tasks Attempted During Meeting:**

Each member is assumed to participate in the SCRUM meeting and contribute to the completion of the SCRUM report and reflections. Since the SCRUM meeting will not take more than 20-30 minutes, there is lots of time left to undertake some of the actual work tasks. In the table below, each member should list what they did to complete the SCRUM report, the reflections, and 1-4 other tasks they completed during the class period. If a task could not be completed, the student should indicate why this was not possible.

|  |  |  |  |
| --- | --- | --- | --- |
| Member | Task Attempted | Time Spent | Complete? |
| Joon | **Reflection q2** |  | **y** |
| Doris | **Reflection q1** |  |  |
| Xiaopeng | **Reflection q4** |  |  |
| Heqing | **Reflection q3** |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**SCRUM Tasks Selected for Next Week**:

The tasks each member has selected to pursue for this class or the next week.

|  |  |
| --- | --- |
| Group Member | Task Description |
| All | Final testing report |
| All | Respective reflection questions |
| All | Try to get code to fully work by debugging current functions that do not pass their test suites |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

This is where you should highlight the major accomplishments of the class.

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Clarification on final test report | **Can make a detailed test report of all test cases** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Things That Went Well in This Meeting:**

Here you can highlight things which worked well. This indicates that the way you worked on these items is working and should be continued.

|  |  |
| --- | --- |
| Topic/Work Item | Reason for Success |
| Distribution of tasks | **All members willingly taking on tasks** |
| Easy test report | **All members were responsible for their test suites throughout these milestones making dividing the test report easy** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Things That Did NOT go Well in This Meeting:**

This is where you can list things which did not go well in the class. You should analyze why this happened and suggest how you can improve it next time. This will lead to the goal of *continuous process improvement*.

|  |  |
| --- | --- |
| Topic/Work Item | Reason for Problem and How to do Better |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

Answer the following questions using your own words. Make sure that each answer comprises a minimum of 100 words.

1. Although we wrote a report on the testing that shows which tests were run and passed or failed, we also updated the function test matrix. What are the advantages of updating the function test matrix in addition to writing the test report?  
     
   Updating the traceability matrix in addition to the test report help us visually track the status of all of the tests, as well as checking the test coverage. We would be able to see which function or components have been tested, what bugs are found and whether the bugs are fixed or not. It give a wholistic overview of our testing progress and making it easy for every team member to grasp the status and progress. It also helps to facilitation communication and collaboration in the sense that team members are aware of the tasks outstanding, and it’s easier for team leader to delegate testing and debugging tasks among the team members. Lastly, it helps our team to stay organized and have a comprehensive documentation of the process and testing efforts we completed.
2. Teamwork on a project like this is vital to success. How well did your team work? If it worked well, what contributed to its success? If it did not work well, what contributed to the problems?  
     
   Our team worked very well throughout the course of these milestones. The biggest contributing factor to the success of this project was the willingness of each member to be responsible for their own tasks. A member was to implement their own functions and we trusted each other in doing so. We saw each other’s test cases and studied further the implementation of each function. Our weekly SCRUM meetings solidified what each of us need to be doing as well as Jira. Jira was managed by one of our team members and we were expected to update our progress using Jira as we moved along in the milestones. This helped to streamline the workflow instead of asking each other where we each member is at in terms of implementation. We were instead able to visually track each other’s progress with Github and Jira.
3. In every milestone you were asked what worked and did not work along the way. Were you able to incorporate what you learned to improving your team’s performance on the next milestone? Did your team learn from its mistakes and improve? If so, why? If not, why?

Yes, I can use what I learned and what I can contribute to my team. And, I constantly learn from our other team members and incorporate this knowledge into refining my responses and abilities. Each of our team members is constantly learned skills and technology from professor and other team members. We also share our experiences with each other and find out mistakes.

Mistakes are inevitable in any team project, but they can also be valuable opportunities for learning and improvement. However, to make the most of your mistakes, you need to have a systematic and constructive way of identifying, analyzing, and resolving them. So, Our team got improvement from complete this milestone project.

1. Did you end up testing the code to the point where you were convinced it worked correctly? Were there any tests that had not passed at the end? If so, what was the impact of this on the project?

Yes. We provide detailed unit testing and integration testing examples that cover testing of multiple functions to ensure that all parts of the software work as expected. Such as Truck Fullness Comparison, Package Validation, Map Functionality, Route Comparison, Package Addition to Trucks, Valid Input Integration, Truck Fullness Integration. Typically, in the final stages of software development, all unit and integration tests should pass to ensure code quality and functional integrity. We currently have no failed tests.