# SFT221 SCRUM Report and Reflections

This report should be completed in the class and submitted at the end of class. Late submissions cannot be accepted without prior approval of the instructor.

**GROUP**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1. Leo Ru | 4. Frank Fu |
| 2. WaiSun Lam | 5. Kam Chun Stanley Tung |
| 3. Xinyang Ma | 6. |

## Milestone 5 Tasks

In this milestone, you should write, implement, and execute integration tests. Integration tests test how multiple functions work together to complete a task. Depending on what is being tested, you might be able to write unit tests to do the testing and automatically compare the results. In other cases, you might need to manually check the output to check it. This will all be stated in the tests where it discusses how they should be run.

As you update the function-test matrix, you will need to add a very brief description for each integration test so the matrix will clearly show what the tests are testing. Acceptance tests will be tested against actual user requirements and will list all the tests for each requirement.

Acceptance tests are the final tests and are largely aimed at showing the customer that the correct output is produced for different inputs. This will largely require manual testing.

**Deliverables Due at end of Lab:**

* Completed SCRUM report and reflections

**Deliverables Due at 23:59 12 Days after Lab:**

* integration tests written and stored in repository,
* integration tests written (store in repo), executed (results in Jira and in test documents) and debugged.
* acceptance tests written and stored in repository.
* Updated function-integration-requirements-test matrix stored to the repository.

**Rubric**

|  |  |  |
| --- | --- | --- |
| Individual | Group Participation | 75% |
| Teamwork | 10% |
| SCRUM Report and reflections | 15% |
| Group | integration tests (well-designed, written and documented) | 20% |
| acceptance tests (well-designed, written and documented) | 20% |
| Test Execution (performed, results recorded, issues created) | 15% |
| Debugging (Bugs fixed, documented, Jira updated) | 5% |
| Function-test matrix updated | 5% |
| Git Usage (used properly with good structure) | 5% |
| Jira Usage (creates issues, tracks progress) | 5% |
| Meets Deadlines | 5% |
| SCRUM Report and Reflections | 20% |

**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** |
| Frank Fu | **Acceptance tests** |  |
| Leo Ru  WaiSun Lam  Xinyang Ma | **Integration tests , test execution, debugging** |  |
| Kam Chun Stanley Tung | **Reflection, matrix** |  |
| Leo Ru  WaiSun Lam  Xinyang Ma  Kam Chun Stanley Tung  Frank Fu | **Scrum report** |  |
| Xinyang Ma | **Update Jira** |  |
|  |  |  |
|  |  |  |

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** |  |
| **Reason for delay or block** |  |
| **Impact on Project** |  |
| **Solution or work-around** |  |
|  |  |
| **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 |
| Integration test | **Discussed/reviewed and suggested integration testing** | **approved** |
| Acceptation test | **Discussed/reviewed and suggested acceptation testing** | **approved** |
| Reflection | **Reviewed and discussed** | **approved** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Integration and acceptation | Test and make sure the program run successfully |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| team | **Integration and acceptation** | **30** | **YES** |
| team | **Scrum report** | **30** | **Yes** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| team | Fix and find bugs |
| team | Run the acceptation test |
| team | Final report |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Tasks assigned for next milestone/ Completion of scrum report 5 | **Everyone on the same page and knows what needs to be done** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| quickly decided on division of labor without conflict | current phase of assignment has clearly outlined deliverables that made it easy for group members to understand the necessary tasks as well as likely workloads involved |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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**:

1. At this point, you are using the GIT hook to automate testing. Have you found that any of the tests failed and prevented you from pushing your code to the repository? If so, how did you handle the situation?

Yes, I have faced situations where some of the tests failed and prevented me from pushing my code to the repository. To solve this issue, I examined the error messages to understand what caused the failure. I went through my codes line by line to find bugs of my codes. I uploaded my files to the repository again after the bugs were fixed.

1. Explain why we are automating the testing process and what the advantages of this automation are.

Automating the testing process has many advantages.

First, automated tests can be run quickly and frequently, increasing productivity of the overall testing process.

Secondly, they facilitate codes reuse. Many codes that have been used in black box or white box tests in one project can be reused in other projects.

Thirdly, they reduce risk of human error in repetitive errors. Humans may lose their concentration while doing repetitive testing, but automated tests do not have this problem.

Furthermore, automated tests can be performed virtually 24/7, providing quick feedback to developers at any time.

1. Did you find the integration and acceptance tests more difficult to write than the black box and white box tests? If so, why were they harder to write? Did you write more white box and black box tests or more integration and acceptance tests?

Yes, integration and acceptance tests are more difficult to write than black box and white box tests because integration tests require deep understanding of how different components interact with each other. In case a test result is different from an expected output, we need to examine each component of the integration test to find the reason and redesign the integration test. Acceptance tests, on the other hand, cover a large scope of functionality in each test case. Each test case is based on business requirements and end-user needs, which can be complex and multi-dimensional.

1. Explain why it is necessary to write integration and acceptance tests given that all of the code has already passed black box and white box tests.

Integration and acceptance tests are necessary because:

Integration tests ensure each individual units of code that have passed black box and white box tests work properly when combined. Even if each individual function works correctly, they may conflict with each other when integrated and tested together. These conflicts can be due to reasons such as incorrect assumptions about data formats or misunderstanding about the intended use of each function. Integration tests help uncover this integrating issue.

Acceptance tests are high-level tests that evaluate the program’s compliance with business requirements and end-user needs. Even if each individual unit of code functions properly, the program as a whole may be meet end-user’s needs and expectation. For example, the program may be too difficult to use. Acceptance tests can validate the final product align with the user expectation.