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

**GROUP**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

|  |  |
| --- | --- |
| 1.Wilson Sum | 4. Samin Sorayya |
| 2.Sasawat Yimleang | 5.Radmehr Behzadfar |
| 3.Lebna Noori | 6. |

## Milestone 4 Tasks

**Deliverables Due at end of Lab:**

* Completed SCRUM report and reflections

**Deliverables Due at 23:59 6 Days after Lab:**

* Implemented Functions
* Implemented blackbox tests (store in repo), executed (results in Jira and on corresponding test documents) and debugged,
* whitebox tests written and stored in repository.
* whitebox tests implemented (store in repo), executed (results in Jira and on corresponding test documents) and debugged.
* Updated function-test matrix stored in the repository.
* Completed hook for test automation

**Rubric**

|  |  |  |
| --- | --- | --- |
| Individual | Group Participation | 75% |
| Teamwork | 5% |
| SCRUM Report | 10% |
| Automation Hook | 10% |
| Group | Implemented Functions (well-designed, written and documented) | 20% |
| Whitebox tests (well-designed, written and documented) | 20% |
| Test Execution (performed, results recorded, issues created) | 20% |
| Debugging (Bugs fixed, documented, Jira 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** |
| **Wilson Sum** | **Scrum report Fill out, “summary of meeting”, ”summary of decisions made”, personal Jira management. Made unit tests for addPointToRoute functions and addroute in conjunction with all getRoute functions. Debugged addpointToRoute, debugged addRoute and debugged validateVolume functions.** |  |
| **Lebna Noori** | * **Analyze the instruction, task, and the obstacle of the project.** * **Completed the SCRUM report.** * **Modified ShortedPath and made unit test and debugged as well.** * **All reflection questions.** | **none** |
| **Samin Sorayya** | * **Participated in the group in-person meeting.** * **Filled out the Scrum report.** * **Utilized Jira management.** * **Implemented functions:**    + **validateLocation**   + **findAvailableTruck** * **Created unit tests for functions:**    + **validateLocation**   + **findAvailableTruck** * **Debugged:**    + **validateLocation**   + **findAvailableTruck** |  |
| **Sasawat Yimleang** | * **Scrum report Fill out, “Major outcome”, ”Things went well”, “Things did not go well”** * **Debug validateVolumn and calculateLimitFactor functions.** * **Create Black Box and White Box test cases for validateWeight, validateBoxSize, validateVolumn and calculateLimitFactor functions.** * **Create getUserInput function.** |  |
| **Radmehr Behzadfar** | **Participating in inperson meeting of group**  **Understanding the difficulties, challenges and duties**  **Creating test cases for “getPossibleMoves” and “distance” functions**  **Creating Unit tests**  **Test the functionality of Tests using both visual studio and prompt command**  **Filling both test traceability and test report**  **Teaching some other members the usage of local unit testing**  **Refining the codes**  **Updating the situation on jira**  **Committing and pushing the data and test which was obligated** |  |
|  |  |  |
|  |  |  |

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 |
| Reviewed how to do unit test cases | **Wilson went over how to do the unit test with all the members.** | **Everyone now should know how to make a unit test. Samin filmed demonstration and distributed amongst members.** |
| Dividing functions to test/implement/debug | **Functions made by team mates and provided by the professor were divided amongst the members to test, implement and debug.** | **Everyone now has a fair share of work and is in charge of their functions.** |
| Divide scrum report tasks | **Divided sections to fill out amongst members.** | **All members are to fill out their own individual parts, Lebna was assigned the reflection questions but in return she only must test and debug one hard function. Leaders Sasawat and Wilson filled out parts specifically on meeting.** |
| Explained bug reporting | **Went over how to report bug and to also use matrix traceability report.** | **Leader Sasawat went over what is a matrix traceability report and explained how to use it.** |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Changed workload from testers and coders to hybrids. | The number of functions and complications of the functions being tested are to be distributed amongst those who implement and test them. Individual members will be fully in charge of their own assigned functions meaning they must implement, test and debug their own program. This helps because the functions are complicated and will take time to relearn if it is split in to two workers. |
| Divided Scrum report | As usual, each week everyone must fill out their own individual parts of the scrum report. Though some of us were given more parts, but in exchange leverage over other parts were downsized so that said members workload won’t be that much. Leaders as usual must fill in the meetings part since they are in charge of overseeing the meeting. |
| Everyone reviews how to do a unit test | All members must review how to do a unit test because they are all in charge of testing their own functions therefore they must know how to run a unit test. |
|  |  |
|  |  |
|  |  |
|  |  |

**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? |
| Wilson Sum | **Teach everyone how to make unit tests. Filled out parts for scrum report. Set up personal JIRA** | **30 minutes** | **Yes** |
| Sasawat Yimleang | * **Analyze the instruction, task, and the obstacle of the project.** * **Completed the SCRUM report by updating the outcome, things went well, and things did not go well at the meeting.** * **Create the White box test cases for validateBoxSize function.** | **90 mins** | **Need to continue create more test cases.** |
| Lebna Noori | * **Analyze the instruction, task, and the obstacle of the project.** * **Completed the SCRUM report.** * **Created ShortedPath function implementation and testing and debugging as well.** * **All reflection questions.** | **Func (2 hrs)**  **Reflections (40 mins)** | **Need time to debug the test cases** |
| Radmehr Behzadfar | **Fully understanding of how to use preexisted unit testing and how to configure and set it with our personal computer.** | **90 min** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 |
| Wilson Sum | ScrumReport fill out, integration of own funcs debugged, fill in matrix traceability report for own funcs made, |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Analyzed the assignment of MS4. | **A clear understanding of the requirement and goal of MS4 will make members understand their responsibility.** |
| Assigned responsibilities on testing part to each member. | **Each member is responsible for the test cases and debugging the function below:**   * **Radmehr: getpossiblemoves and distance functions.** * **Samin: validatelocation, findavailabletruck and validatemapsize functions.** * **Lebna: shortestpath function and the reflection questions.** * **Sasawat: getuserinput and getclosestpoint functions.** * **Wilson: addpointtoroute, printmap, getblueroute, getyellowroute, getgreenroute and addRoute functions.** |
| Declared the due date of each part. | **The due date for all tasks in this Milestone is on Sunday in case that has the unexpected happens. This will ensure the project will go properly.** |
|  |  |

**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 |
| Understanding of the assignment of MS4. | **The communication channel between members allowed for discussion between members immediately and making sure everyone is on the right track.** |
| Task assignment | **We discussed and made agreements about the person responsible for each task.**  **The meeting ensured that each task was allocated to a responsible team member.**  **Clear roles and responsibilities were defined, avoiding confusion or duplication of efforts.** |
| Participation | **All Team members presented actively participated and contributed to the discussions.**  **Active participation fostered collaboration and helped generate valuable input and solutions.** |
| Decision-Making | **Decisions were made in a timely and collaborative manner.**  **The decision-making process was efficient and did not unnecessarily prolong the meeting.** |
|  |  |
|  |  |
|  |  |

**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 |
| One member must leave the group. | **Mostafa has the emergency issue at his home country, so he needs to leave the group for now.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Reflections**:

1. After you run your blackbox and whitebox tests you are asked to record the results in both the original test document as well as in Jira. Explain why it is a good idea to record the results in both places.
   1. **Complete Documentation**: By storing information from tests in both locations, it is guaranteed that all test results are properly documented and readily available.
   2. **Traceability**: By connecting test cases in the test document with appropriate issues in Jira, a direct link is made between the tasks or bugs that the tests are meant to fix.
   3. **Collaboration and communication**: Comparing the findings from the two locations encourages communication and collaboration among team members who might favour different approaches.
   4. **Easy Reporting**:  Information from both sources can be used to create reports and analyze metrics, giving important information into the testing process.
   5. **Backup**: By storing results in two locations, you can be sure that important test data won't be lost in case of data loss or other problems in one system.
2. Why did we wait until the fourth milestone to write the whitebox tests?
   1. In my opinion we waited until milestone 4 because white box so that we would have a stable codebase, and make necessary changes or fixes to have a solid code base for white-box testing based on black-box testing.

For example:

* + 1. **Testing Priority**: Black-box testing is the main area of concentration throughout the first phases of any software development project which verifies the functionality of the product from the viewpoint of end users. Delivering a high-quality product requires making sure the program satisfies the requirements it was designed to meet and behaves as expected by users.
    2. **Bug fixes**: Many of the major bugs found during black-box testing were fixed by the time the fourth milestone is achieved. It is important to take care of these high-priority issues early to make the program more reliable and lower the possibility that white-box testing execution may be slowed by unsolved bugs.
    3. **Efficiency and Focus:** By delaying the start of white box testing until the codebase has fixed, testers can focus on a more developed and cleaner codebase. By minimizing the need for ongoing testing and rewriting caused by frequent code changes, this strategy maximizes testing efforts and executions.
    4. **Clear Functional Requirements**: The early parts of milestones play an important role in defining clear functional requirements, and knowing what is required. A clear set of requirements provides a strong basis for both black-box and white-box testing, instructing testers on which elements of the software to concentrate on during each stage.

Black-box testing should be prioritized early on in the project to guarantee efficient use of resources. Black-box testing often has a wider focus and calls for more testing resources.

1. For a given function did you produce more Blackbox or Whitebox tests? Explain why your answer (more Blackbox or more Whitebox) happens for most functions.
   1. Checking the internal code logic, routes, and sections of the function being tested is known as white-box testing. I used the function's specification in the test I prepared, but I also took into consideration how it was internally implemented and behaved.

In the shortestPath test case for the main function, I call the function with certain start and destination points and check the accuracy of the returned route using the specified map and points. This strategy involves taking note of the core workings of the function, such as how it uses the map to determine the shortest route and steer clear of structures. As a result, the shortestPath test case can be defined as a white-box test since it looks into the internal behavior of the function and makes use of that information to create test cases. White-box testing is helpful for validating that code paths are correct and that the function follows its implementation reasoning.

1. Explain the purpose of the automation hook for GIT and explain how it can improve the quality of the software in the project.
   1. Git's automation hooks were created to increase software quality by automating routine tasks, keeping consistency, and removing common mistakes. Here are a few ways automation hooks can raise the quality of software:
      1. **Setting Coding Standards**: During commits, automation hooks provide automatic checks for coding standards, naming conventions, and code formatting. This improves readability and maintainability of code and ensures developers use consistent coding practices.
      2. **Automated Test Execution**: With the help of automation hooks, automated tests, including unit tests and integration tests, can be started before code is committed or pushed. This active testing strategy guarantees that code updates connect easily with the existing codebase, cuts out regressions, and helps with the beginning detection of errors.
      3. Automation hooks can work with security tools to search the codebase for existing errors and security errors, preventing problems. The project's safety record is greatly improved by automatically identifying and fixing any issues.
      4. **Preventing Bad Commit Behaviour**: Automation hooks can set commit message standards, requiring programmers to include informative and detailed commit statements. They can also link commits to particular issues or tickets, which makes it simpler to track changes and follow the development process.
      5. **Direct Commits to Specific Branches Are Prevented**:  Automation hooks stop direct commits to important branches by introducing branch protection rules. This minimises the possibility of introducing bugs or conflicts in important parts by making sure all code changes go through difficult code reviews and testing before merging.

Overall, it is all about efficiency and uniformity, and In general, automation hooks increase code quality, decrease human mistake. An effective and productive development workflow is done by automating routine tasks so that developers can concentrate more on writing code and less on following manual processes.