# **MILESTONE 3** -- 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**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Members Present**:

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

## Milestone 3 Tasks

In this milestone you will create issues to design the functions, design all of the functions you need to complete the project and store the specifications in the repository. As soon as the specifications start to be produced, you can start to design the blackbox tests (what they test, how to perform them and test data). Once tests are written, they can be implemented and added to the repository and any team members not otherwise busy can start to implement the functions. You will also build a function-test matrix that shows the blackbox tests for each function. This will be maintained through the testing cycle as new tests are added.

**Deliverables Due at end of Lab:**

* Completed SCRUM report and reflections

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

* A set of function specifications stored in the repository,
* A set of blackbox tests as test documents with test data for the functions.
* Start writing blackbox test code and store in repository. (at least 1 required)
* Start implementing functions and store in repository. (optional)
* A function-test matrix added to the repository.
* Updated Jira project to show activities and progress.

**Rubric**

|  |  |  |
| --- | --- | --- |
| Individual | Group Participation | 75% |
| Teamwork | 10% |
| SCRUM Report | 15% |
| Group | Function Specs (documented, correct, complete, well-written) | 20% |
| Test documents (well-written, complete, good test data) | 20% |
| Test Code (well-designed, written and documented) | 10% |
| Git Usage (used properly with good structure) | 5% |
| Jira Usage (creates issues, tracks progress) | 10% |
| Meets Deadlines | 10% |
| SCRUM report & reflections | 25% |

**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** | **Filled in “Summary of Meeting:”,” Summary of Decisions Made:**  **” and individual self parts of scrum report. Create unit tests for validateVolume, validateWeight, calculateLimitingFactors.** | **Must still make unitTest for validateDestination awaiting completion(optional since 2 unitTests already made)** |
| **Lebna Noori** | * **individual self-parts of scrum report** * **Answered reflection questions.** * **Applied agile approach** | **No delayed tasks** |
| **Sasawat Yimleang** | * **Filled in “Major outcome”, “Things went well” and “Things did not go well” of the scrum report.** * **Created unit tests for validateSize function.** * **Filled the Traceability Matrix.** * **Created the Test Data Reports.** | **No delayed tasks** |
| **Samin Sorayya** | * **Added a new function prototype in the validate.h header file.** * **Started working on creating test cases for the new function.** * **Continued to focus on completing test cases, and descriptions, and implementing functions that have been started by others.** * **Completed the Scrum report** * **Filled in “Testing Strategy Worksheet”** * **report** | **No delayed tasks** |
| **Mostafa H.** | * **Developed 7 function prototypes.** * **Developed 4 function implementations.** * **Created and completed function specification document report and also provided specifications as comments above each function in header file.** * **Developed header, c file and main files ready for testing process.** * **Created and completed function traceability matrix.** * **Worked and edited the Black box test data report. (Added test type column and edited the final results by comparing unit test results)** * **Edited “Testing Strategy Worksheet”**   **report**   * **Completed and did the final editing of scrum report (developing and editing initial reflection questions, completed meeting tables).** * **Updated Jira** * **Attended all meetings.** * **Committed changes to Github.** * **Did the final editing of deliverables** | **N/A** |
| **Radmehr Behzadfar** | * **individual self-parts of scrum report** * **updating jira** * **over checking the results of the test Cases via test unit** * **committing and pushing the changes to github** | **N/A** |
|  |  |  |

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** | **Unit test for validateDestination Function** |
| **Reason for delay or block** | **Has not or may not be implemented.** |
| **Impact on Project** | **Since only one unit test is required but I did two should not be that bad.** |
| **Solution or work-around** | **Either leave it or find a new function to test.** |
|  |  |
| **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 |
| Discussion of the Assignment | **The team discussed about what the assignment is asking and how they should approach the assignment** | **The assignment is divided mainly into three core parts such as the functions to code, the tests and the weekly scrum report. Within those are even more smaller tasks and each member oversees their corresponding tasks.** |
| Explanation of function creation and testing. | **Briefly explained why we need to create functions, how to do that, the overall testing process at current stage.** | **Group members now have a general idea on how to develop functions and how to delve into the testing process of functions** |
| Assigning Parts | **Two main teams will be required, one being the test team in charge of making test cases, unit tests, testing the functions and making the traceability matrix. Then for the functions themselves, another sub team will be made where they will oversee making the functions and reporting the specs of the functions. All work will be then reviewed by everyone and necessary changes need to be applied.** | **Sasawat and Wilson will be doing the tests with Sasawat making the test cases traceability matrix and Wilson making the unit tests to test the functions. Samin, Radmehr and Mostafa will oversee figuring out what functions will be necessary to be made and to also implement them as well record their specs.** |
| Jira/Git Responsibilities | **As of now, since everyone knows how to use Jira and Git, everyone will oversee making their own Jira tasks and committing their own work to the Group Repository** | **Each member will be in charge from now on for all their own tasks written on Jira as well as committing to the group repos.** |
| Project deliverables | **Discussed about the specific deliverables.** | **The disagreement was on traceability matrix table. Decided to create both function and test cases traceability matrix.** |
|  |  |  |
|  |  |  |
|  |  |  |

**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 |
| Sub Teams formed | Breaking the group into sub teams will allow the core aspects of the assignment to be met by responsible memebers. |
| Individual Jira Management | The load work for the team leaders will be lessened so they may focus on their parts. Also, since each member now knows how to use Jira they must be in charge of keeping themselves on course for the assignment. |
| Test methods team will develop the test data | Test team will develop the test data necessary to test the positive test cases for the function and the negative test cases to test all the outer bound entries. |
| Function developing | Mostafa would be the lead of the team of function development. The functions are necessary so that the program will be able to work properly and meet the business requirements. |
| Online meeting | We decided to give group members enough time to fully grasp the idea of the milestone and analyze the project, before meeting online via Zoom application and discuss various aspects of the milestone. |
|  |  |

**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 | * **Analyze the instruction, task, and the obstacle of the project.** * **Thinking up Test data and practicing unit test making.** | **1 hr** | **no** |
| 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 of the meeting.** * **Declare the due date of each task to each member.** * **Generate test concepts and test plans with Wilson.** | **80 mins** | **Yes** |
| Lebna Noori | * **Analyze the instruction, task, and the obstacle of the project.** * **Completed individual self-parts of scrum report.** * **Completed Reflection Questions** * **Analyzed the codes process, test cases, and instructions for reflection questions.** | **120 mins** | **Yes** |
| Mostafa | * **Explained the milestone to other group members.** * **Analyzed the specification of MS3 in the meeting.** * **Talked about changes in Repos structure need to be made.** * **Assigned different parts of milestone to members based on their strength and interests.** * **Explained the functions I developed in last week to other members and advised the unit test responsible member which function to work on.** | **80 min** | **Yes** |
| Radmehr Behzadfar | * **Analyze the instruction, task, and the obstacle of the project.** * **Completed individual self-parts of scrum report.** * **Being explained about the rule in the group** |  |  |

**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 | Scrum Report, Self managing of Jira, Debugging and recoding funcs (unless team mates wish to remain in subset groups) |
| Samin Sorayya | * Implement functions that have already been started by others, ensuring proper functionality and adherence to specifications. * Conduct blackbox testing for the implemented functions, focusing on testing the system's behavior based on requirements and expected outcomes. * Conduct whitebox testing for the implemented functions, considering internal implementation details and code structure to ensure thorough coverage and identify any potential issues or bugs. * Collaborate with the test team to review and refine existing test cases, and create new test cases as needed. * Prepare and finalize the SCRUM report, including documenting completed tasks, reflections, and any relevant updates or changes to the project. * Stay updated with the progress of the project, actively participate in team discussions, and contribute to decision-making processes. |
| Mostafa | * Develop implementation code of functions. * Edit implementation of functions that are already developed based on the results of unit tests. * Review and edit test cases / create new test cases if needed. * Complete and do the final editing of the scrum report. * Updating Jira * Committing changes to Git repos |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Analyzed the assignment of MS3. | **A clear understanding of the requirement and goal of MS3 will make members understand their responsibility.** |
| Assigned roles and responsibilities on coding and testing part. | **The validation functions will be the work of Mostafa, Radmehr, and Samin. The test case code and related documentation will be the work of Sasawat and Wilson. This assignment will reduce workload and redundancy.** |
| Declared the due date of each part. | **The due date will make the project workflow more efficient.** |
| Agreed upon project deliverables | **We achieved clear expectations and goals for the project.** |
| Online meeting settings | **Prioritize the analyzing the problem before trying to work would lead to achieve a better outcome.** |
|  |  |

**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 MS3. | **The communication channel established in the first week between members allowed for discussion between members 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.** |
| Meeting time management | **The meeting adhered to the scheduled timeframe.**  **The efficient use of time allowed for productive discussions and timely decision-making.** |
| Participation | **Team members who 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 |
| The due date. | **Some of the members must work and need extension time to finish the tasks. We decided to change the due date of each part to keep the workflow still efficient as we can.** |
| Task participation | **Some members felt that they are left with no task, and nothing remains for them to contribute.**  **Change in the task allocation and assignment needs to be applied.** |
|  |  |
|  |  |

**Reflections**:

1. In this milestone, we write the blackbox tests but not the whitebox tests. Explain why we can write the blackbox tests but not the whitebox tests.

In this milestone, we write the blackbox tests but not the whitebox tests because the focus is on testing the functionality of the functions from an external perspective without considering the internal implementation details. Blackbox testing is based on the system's requirements and specifications, treating the functions as "black boxes" where the internal workings are unknown. During this phase, we may have only implemented the function prototypes and a limited number of function implementations. Without the complete implementation of all functions and their internal logic, it would be difficult to write comprehensive whitebox tests that specifically target and validate the internal code paths, branches, and algorithms. However, we can still write blackbox tests by examining the expected behavior of the functions based on their inputs and desired outputs. Blackbox tests can be designed to cover different scenarios, edge cases, and requirements to ensure that the functions meet the expected functionality. These tests focus on the external behavior of the functions, independent of their internal implementation details. Once the complete function implementations and internal logic are developed, we can then proceed to write whitebox tests to thoroughly validate and verify the internal code paths, algorithms, and error handling within the functions.

1. Explain why we need the function-test matrix and why it is important in a large project.

The function-test matrix is a tool used to match up test cases with a system's features or functions. It offers a structured and organized technique to guarantee that all system features or activities are thoroughly tested. The function-test matrix plays an important part in a large project for many reasons, for example:

**Coverage of Tests**: Large projects usually include a lot of functions or other features. The clear linking of each function or feature to a particular test case in the function-test matrix helps in guaranteeing complete test coverage. It helps in locating any holes or areas that have not been thoroughly inspected.

**Traceability**: The function-test matrix shows a direct relationship between features or functions and the related test cases. Better test coverage tracking and reporting becomes available by this traceability. It becomes simpler to show that every feature or function has been tested and offers proof of complying with project requirements.

**Planning and Prioritizing** **Tests**: The matrix is a useful tool for planning and prioritizing tests. It is simpler to prioritize testing efforts based on importance, or project schedules by showing the relationship between functions, features and test cases. It guarantees that high-priority functions are properly reviewed and helps in the effective usage of resources.

**Communication and Collaboration**: function-test matrix helps in effective team working together and communicating in large projects with multiple participants. It offers a standard point of discussion for managing testing activities, as it can be used as a shared document by developers, testers, and project managers to make sure that everyone is aware of the test coverage and progress.

**Change Impact:** The function-test matrix becomes a useful tool for change impact analysis as the project grows and changes take place. The matrix helps in locating the related test cases that need to be updated or added when a function or feature is changed. It helps in understanding how changes will affect the test coverage currently in place and makes sure that relevant tests are modified properly.

Overall, the function-test matrix serves as a crucial tool for requirement management, test planning, traceability, collaboration, risk management, and scalability in large projects. It helps ensure that the project stays on track, meets the intended business objectives, and delivers high-quality software solutions.

1. Other life cycle models left team members idle while waiting for parts of the project to be completed. Describe how an agile model, like the one we are using, avoids this problem and keeps the whole team busy all the time. Does this make managing the project simpler or more complex and why?

An agile model, like the one being used, cuts out the problem of team members being idle while waiting for parts of the project to be completed through its continuous and incremental approach. In agile, work is divided into smaller iterations or parts, and each iteration creates a working increment of the project. This technique keeps the entire team motivated and productive throughout the project's stages. Which makes managing the project simpler, but can also additional complexities can be involved. It makes it simpler because:

* Agile supports a continuous workflow by breaking the project down into smaller, more manageable parts. The team switches to the next iteration as soon as one is finished, keeping an ongoing cycle of work. Depending on the current iteration, team members are involved in different kinds of activities such as planning, creating, testing, and review.
* Agile creates mutually beneficial collaboration and communication between teams in a supportive setting. Team members work on multiple parts of the project at once, using their individual skills and mutually supporting one another. In this cooperative setting, there is less downtime because team members can help add to different parts of the project.
* Flexibility and Adaptability: Throughout the project, Agile allows for flexibility and adaptability. The team may update priorities and incorporate changes into future iterations as requirements change. With this flexibility, the team is continually working on the most significant and relevant projects, reducing downtime from waiting for expired or incomplete needs.
* Multitasking is frequently highlighted in agile techniques, allowing team members to have an additional set of skills. A team member can help in other areas where it is needed after finishing their tasks for a particular iteration. This method of multitasking makes sure that team members can do different duties, which cuts down on downtime and creates a more flexible and cooperative team.

Agile requires constant participation and involvement from all members of the team. If members of the team refuse to work together it could make the completion of the project much more challenging and time-consuming.

* Prioritization and efficient planning are very important to maintaining an ongoing flow of work and minimizing challenges in an agile project. The team could face several challenges, because of an unclear and effective plan, for example:
* A well-defined plan gives the project direction and establishes how it will proceed. Team members could feel confused or unsure about their jobs and priorities without it. Confusion, delays, and mistakes might happen from this. Without proper planning, sharing resources could become chaotic and result in either an overuse or underuse of team members. Agile projects also frequently depend on one another, so if one mistake is made in one part it will affect all the other parts of the project.