# 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**: C

**Members Present**:

|  |  |
| --- | --- |
| 1.Manav Zadafiya | 4.Fenil Soni |
| 2.Sunny Vavadiya | 5. |
| 3.Ashraf Bharot | 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** |
| **Manav, Fenil** | **Functions Implementation and complete hook for test automation** |  |
| **Ashraf** | **White box testing code and update repository** |  |
| **Sunny** | **Black box testing code and update repository** |  |
| **All members** | **Scrum Report and reflection** |  |
| **Manav** | **Test Matrix Uploaded** | **Delayed from ms3** |
|  |  |  |
|  |  |  |

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** | **Uploading Test Matrix** |
| **Reason for delay or block** | **Forget to upload** |
| **Impact on Project** | **Not much as we did not start testing yet** |
| **Solution or work-around** | **More activity on JIRA and create check list for all milestones** |
|  |  |
| **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 |
| Functions Implementation | **How to implement functions that we have created in ms3 (Shipment.h) last week** | **Implementation Started** |
| JIRA | **Updating activities in JIRA as per milestone 4** | **Completed** |
| Scrum Report | **Scrum Requirements and discuss reflection** | **Scrum Completed** |
| Testing | **White and Black box testing functions** | **Started writing test functions** |
| Test Automation | **How and who will create hook for test automation** | **Task planned and assigned to members** |
| Milestone 5 and 6 | **Analyzing and understanding Next milestones requirements** | **Understood upcoming challenges** |
|  |  |  |

**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 |
| Prioritization of tasks | Equal amount of work assigned to each member of team to ensure productivity and timely completion of assigned task. |
| White Box testing | Need new testing implementation, executed, and recorded in matrix for MS04s to test the program thoroughly. |
| Black Box testing | After completing function implementation, one assigned member start testing functions to ensure smooth flow. |
| Function implementations | Implementation will follow Shipment.h that was developed last week as function specs. |
| understanding the requirements of the test plan and milestone | follow the plan and help the other team member if they face any problem. |
|  |  |
|  |  |

**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.

|  |  |  |  |
| --- | --- | --- | --- |
| ALL | Analysis discussion, Analysis of debugging black box testing that was done last week and discuss about white box implementation and execution. | 1hr | Yes |
| ALL | **Analysis discussion, Analysis Function implementation in program and discussed** | **45hr** | **Yes** |
| ALL | **Scrum report** | **40min** | **Yes** |
| ALL | **Jira and Github Project page updated and assigned** | **25min** | **Yes** |
| ALL | **Discussion on hook automation** | **30min** | **Yes** |
| ALL | **Discussion for next milestone and tasks** | **20min** | **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 |
| Manav | Test matrix and Test files management and Git/Github |
| Fenil | Jira Control and task management and review documentation |
| Ashraf | Integration tests |
| Sunny | Acceptance Tests |
| ALL | Group Meeting on Thursday, 15th Nov. |
| ALL | Scrum report |
| ALL | Test execution |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Major Outcomes of Meeting:**

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

|  |  |
| --- | --- |
| Outcome | Impact on Project |
| Function implementation | **Function implementation was done according to the function specs that was written for last week** |
| Black Box testing | **The team has finished attempting black box testing last week even though it was for this week’s requirement. We have found some bugs in our testing codes, we ticketed on matrix and Jira kanban, Git project(kanban), and issue was resolved** |
| White box testing | **White box testing codes were implemented and executed.** |
| Hook implementation | **Hook implementation was discussed and screenshot was sent to professor.** |
|  |  |
|  |  |
|  |  |

**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 |
| SCRUM | **All contributed.** |
| Git | **Useful for version control and keeping track of changes** |
| Meeting | **All attended meeting and had discussion about milestone progress.** |
|  |  |
|  |  |
|  |  |
|  |  |

**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 |
| N/A |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.

Documenting the test results in both the original test document and Jira offers numerous benefits. The test document serves as a comprehensive historical record, documenting all tests conducted, their objectives, and obtained results, ensuring traceability for future reference. Meanwhile, Jira's integration with the development workflow links test results to specific issues or user stories, enabling easy tracking of bug fixes or feature implementations. This fosters collaboration among team members, including developers, project managers, and product owners, who can access the results for communication and decision-making. Real-time updates and notifications in Jira keep everyone informed about the software's current quality status and potential issues. Additionally, Jira's built-in reporting and metrics capabilities allow for the generation of valuable insights into the software's overall quality and the effectiveness of testing efforts. Finally, having test results stored in both places ensures redundancy and compliance with documentation requirements for audits and projects with strict quality standards.

It also helps with teamwork and communication. Overall, recording test results in both the original test document and Jira helps with documentation, accountability, teamwork, and analyzing past performance. It's a practical way to ensure a smooth testing process and improve the overall quality of the software.

1. Why did we wait until the fourth milestone to write the whitebox tests?

In our project, we were required to delay writing whitebox tests until the fourth milestone. This decision must have been driven by several factors. Initially, we prioritized blackbox tests to validate the software's core functionalities and ensure it met the required criteria. Waiting for the fourth milestone allowed the codebase to stabilize, reducing the risk of test rework due to frequent changes. Moreover, we adopted a risk-based testing approach, addressing high-risk areas with blackbox tests first. After verifying critical parts, we introduced whitebox testing to gain deeper insights into lower-risk areas and improve overall code quality. Prioritizing blackbox testing in the initial stages enables the testing team to achieve broader test coverage and address high-level issues arising from user interactions and external systems. Additionally, adopting an iterative testing approach gradually increases testing complexity as the project progresses, ensuring

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.

For a given function, int checkSize(const double size), I found more test cases by using the black box techniques in this workshop code. However, I believe to adequately test a program’s code, a combination of both testing techniques is required. Because Blackbox testing mainly deals with feeding different inputs to a code and testing its functionality from a user’s perspective who has little or no understanding of how the software works. However, white box testing is more comprehensive and deals with testing how the code works in different scenarios while having a proper understanding of the internal structure of the program and trying to find the potential bugs in the program which might not be caught with black box testing only. However, black-box tests are generally easier and faster to develop than white-box tests as the tester needs to consider only the inputs and outputs of the code without requiring any knowledge of the actual logic and structure of the written code.

1. Explain the purpose of the automation hook for GIT and explain how it can improve the quality of the software in the project.

Automation hooks in GIT are like little helpers that automatically run certain scripts or tasks when specific things happen in the version control system. For example, when someone adds new code or makes changes to the existing code (called code commits) or when they want to add their changes to the main project (called pull requests). These hooks offer lots of advantages. They help with continuous integration and continuous deployment, which means that whenever someone adds or changes code, it automatically goes through a bunch of tests to check if everything is okay. This gives quick feedback to developers about the quality of their code and makes it easier to deploy changes to the live website or app. The hooks also help in maintaining good code quality by running checks on the code to make sure it follows the rules and standards set by the team. They can even catch bugs early on, so developers can fix them before they become big problems. With automation hooks, everyone on the team follows the same rules, which makes the development process more consistent. This way, developers don't have to worry about running tests manually or remembering all the rules because the hooks take care of it. This saves time and lets developers focus on the fun and creative parts of coding. Overall, automation hooks make sure that the code is in good shape, reducing the chances of mistakes or errors, and make it easier to manage and keep an eye on things.