

## MILESTONE 2 -- 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:** \_\_\_\_\_ 4 \_\_\_\_\_

### Members Present:

|                    |                 |
|--------------------|-----------------|
| 1. Hsien-Ting Liao | 4. Shu-Ting Hsu |
| 2. Rong Chen       | 5. Pei-Ti Chen  |
| 3. Yansong Gao     |                 |

### Milestone 2 Tasks

Some of the software for the project has already been written for you and is available on Blackboard. You must use this in your project and every team should add it to the source code for their repository. Anything in the main function is simply for demonstration purposes and can be replaced. The software you are being given has not been tested and you will need to test it.

You need to study the problem and the code provided for you and then:

- Add any new data structures you will require This will require a thorough analysis of the problem and the existing software. This should be done by creating a new header file in the directory where the rest of the source code has been placed. You do not want to go back and modify it later if you can avoid it as it will slow the project.
- Create a test plan for the project by replacing the text in the supplied test plan template with your test plan.

### Deliverables Due at End of Lab

- Completed SCRUM report & reflections

### Deliverables Due within 48 hours of lab

- An analysis of the problem (no written artifacts produced),
- A series of data structures created as header files and stored in the repository,
- A test plan stored in the repository.

### Rubric

|            |   |     |
|------------|---|-----|
| Individual | Group Participation                                   | 75% |
|            | Teamwork  | 10% |
|            | SCRUM Report  | 15% |
| Group      | Data structures (complete, correct and well-designed) | 20% |
|            | Test Plan (complete, well-written)                    | 20% |
|            | Git Usage (used properly with good structure)         | 10% |
|            | Jira Usage (creates issues, tracks progress)          | 10% |
|            | Meets Deadlines                                       | 15% |

|  |                              |     |
|--|------------------------------|-----|
|  | SCRUM Report and 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 |
|-----------------|---------------------------------|-----------------------|
| Hsien-Ting Liao | Create testing plan, manage git | N/A                   |
| Rong Chen       | Create new data structures      | N/A                   |
| Yansong Gao     | Create new data structures      | N/A                   |
| Shu-Ting Hsu    | Create testing plan             | N/A                   |
| Pei-Ti Chen     | SCRUM Report and Reflections    | 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.

|                                  |     |
|----------------------------------|-----|
| <b>Delayed or Blocked Task</b>   | N/A |
| <b>Reason for delay or block</b> |     |
| <b>Impact on Project</b>         |     |
| <b>Solution or work-around</b>   |     |
|                                  |     |
| <b>Delayed or Blocked Task</b>   |     |
| <b>Reason for delay or block</b> |     |
| <b>Impact on Project</b>         |     |
| <b>Solution or work-around</b>   |     |

## Summary of Meeting:

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

| Topic                        | Discussion Summary   | Outcome  |
|------------------------------|--|--|
| Create new data structures   | Create data structures to do the project.  | Create new data structures                                     |
| Create testing plan          | Create a testing plan and think what will cause the errors.  | Create testing plan  |
| SCRUM Report and Reflections | Discusses the advantages of analyzing a problem and design software before starting to do a project. | Learned the concept of analyzing a problem and design software |
|                              |  |  |
|                              |  |  |
|                              |  |  |
|                              |  |  |

## 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  |
|------------------------------|--|
| Create new data structures   | Need to know which structures we need in this project and the prototype of the function.                                 |
| Create testing plan          | Need to know what user will input and need to check the boundary before release.   |
| SCRUM Report and Reflections | The last part in this week's meeting, because need the outcome for create new data structures and testing plan to do it. |
|                              |  |
|                              |  |
|                              |  |
|                              |  |

## 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? |
|-----------------|---------------------------------|------------|-----------|
| Hsien-Ting Liao | Create testing plan, manage git | 3 hours    | Yes       |
| Rong Chen       | Create new data structures      | 3 hours    | Yes       |
| Yansong Gao     | Create new data structures      | 3 hours    | Yes       |
| Shu-Ting Hsu    | Create testing plan             | 3 hours    | Yes       |
| Pei-Ti Chen     | SCRUM Report and Reflections    | 3 hours    | 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    |
|-----------------|---------------------|
| Hsien-Ting Liao | Testing, manage git |
| Rong Chen       | Make function       |
| Yansong Gao     | Make function       |
| Shu-Ting Hsu    | Testing             |
| Pei-Ti Chen     | Debug               |
|                 |                     |
|                 |                     |
|                 |                     |
|                 |                     |
|                 |                     |
|                 |                     |
|                 |                     |

### Major Outcomes of Meeting:

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

| Outcome                      | Impact on Project |
|------------------------------|-------------------|
| Create new data structures   | Finished M2       |
| Create testing plan          | Finished M2       |
| SCRUM Report and Reflections | Finished M2       |
|                              |                   |
|                              |                   |
|                              |                   |
|                              |                   |

### 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  |
|------------------------------|---|
| Create new data structures   | With teamwork, everyone focuses on what we need to do and finish it. We did a good job. |
| Create testing plan          | With teamwork, everyone focuses on what we need to do and finish it. We did a good job. |
| SCRUM Report and Reflections | With teamwork, everyone focuses on what we need to do and finish it. We did a good job. |
|                              |   |
|                              |   |
|                              |   |
|                              |   |

### 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. In this milestone you have been asked to analyze a problem and design software(functions) to complete the solution without actually writing the software.
  - a. Is this process more difficult than just writing the software to complete the project? If so, why is it more difficult? If not, why is it easier than just writing the software?

When it comes to the software development process, the initial analysis and design phases are often considered more challenging than actually writing the code. This is because these early stages require careful thought and planning. The developers need to think very carefully and clearly about what they need to do and how to do it. They also need to think about what may happen and cause the project not to work. So, if the developers do well in the first step, then the rest of the work will become easier and smoother.

- b. Describe two advantages of developing software in this manner rather than just moving on to writing the functions without writing specifications first.
    1. Reduced Errors: Detailed specifications minimize misunderstandings and errors.
    2. Improved Efficiency: Clear specifications enhance productivity and teamwork, making the development process smoother.Emphasizing analysis and design before writing code reduces errors, rework, and misinterpretations, resulting in a more efficient and productive software development process. It ultimately leads to the creation of higher-quality software.

2. Why is it a good idea to create a test plan? Describe at least 3 advantages of test plans.

Without a test plan, developers don't know what the boundary for the code is and don't know when it will cause an error when users use the project. With a test plan, which contributes to more effective and higher-quality testing processes,

1. Structured Testing: Test plans ensure a systematic and comprehensive approach to testing.
2. Clear Communication: They facilitate communication among project stakeholders, reducing misunderstandings.
3. Risk Management: Test plans help identify and mitigate potential issues early in the development cycle, improving software quality.

3. Describe the process you used to analyze and understand the existing software.

To analyze and understand existing software, a structured process is typically followed.

1. Gather Documentation: Collect all available documentation related to the software. This is the first step a developer needs to take to fully understand what the project asks. Also, this is what we are doing for milestones 2.
2. Visualize the System: Create visual representations of the software, such as flowcharts, data flow diagrams, or pseudo code. These can help developers understand the project more clearly.
3. Use testing and debugging: Running the software and systematically testing different features, inputs, and scenarios can help understand its current state and identify issues. And this is why we need QA to test a project before its release.
4. Assess Performance: Evaluate the software's performance to identify bottlenecks or areas that may need optimization.
5. Feedback: Share your findings and recommendations to ensure any necessary changes or improvements.