University of the Witwatersrand, Johannesburg
School of Computer Science and Applied Mathematics

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

Sprint 4 Instructions and Rubric

1 Instructions

Sprint 4 is due on the week of the 29th of May. By that time, you should have a largely complete and polished product.

As usual, make sure you have contacted your marker to mark sprint 3, and have contacted your client to do a sprint planning meeting for sprint 4. It is your responsibility to organise these meetings.

Remember that if anything your client says contradicts these instructions, your priority is to follow these instructions. Feel free to contact me should you have questions.

At the end of the sprint, you will be marked in a similar fashion to the previous sprints. Your client will also be completing a survey after your sprint 4 review meeting, in which they declare how happy they were with your project and whether or not it met their requirements, which will contribute to your project mark.

Note: In your demos, you should be using the final release of your sprint. Anyone in the group with access to an appropriate device (e.g. a browser if your software is a webapp, an Android phone or emulator if it is aimed at Android mobile devices, etc.) should be able to demo the project.

1.1 Scrum Methodology

As with sprint 3, you will need to follow the scrum methodology. Proof of daily scrum, sprint planning, sprint review, backlog refinement and sprint retrospective meetings must be provided. Make sure you do a sprint review at the end of the sprint, as your client needs to see the finished product in order to fill in their survey.

1.2 Requirements Engineering

Same as sprint 3. Ensure you continue documenting for sprint 4.

1.3 Software Architecture

For sprint 4, you need to ensure that your existing UML architecture diagrams are up-to-date. For example, your use case diagram needs to contain all your sprint 4 user stories.

1.4 Software Implementation

By the end of the sprint you should have completed at least four user stories, in addition to those completed in sprints 1-3. A user story is complete if it passes its associated User Acceptance Test (see below).

At this point, if your project is supposed to have a distributed architecture (e.g. client-server), I will not accept local or dummy databases. Consider using your VMs.

Your project should be a viable product. It is fine if you have not implemented "nice to have" features, but the core functionality must be present.

1.5 Continuous Integration

Same as sprint 3. Make sure you make a release for sprint 4.

1.6 Test-Driven Development

Same as sprint 3, although for full marks you need to achieve 80% coverage with unit tests (you may ignore UI code). Your unit tests need to be comprehensive (e.g. comprehensive equivalence/boundary testing).

1.7 Project Polish

A greater emphasis will be placed on the polish of your project. Ensure that there are no bugs, the software looks good and the user experience is good. Remove any placeholders in the software, and make sure that connection or performance issues are sorted out.

Your code should be well-commented and organised logically. Your marker should be able to pick a random file and see good comments explaining the code. External documentation, while not strictly necessary, would be incredibly helpful for this mark.

2 Rubric

	Poor	**Acceptable**	**Good**	**Excellent**	**Weight**
Scrum Methodology	Proof of 1 or 2 meetings	Proof of 3 meetings	Proof of 4 meetings	Proof of all 5 meetings	10
Requirements Engineering	Documentation created, user stories present for sprint 4	Correct format for user stories, product backlog maintained	User stories divided into tasks	All tasks assigned and statuses tracked	10
Software Architecture	1 or 2 viewpoints described, very poor UML diagrams	3 or 4 viewpoints described, some UML mistakes present	All viewpoints described, few UML mistakes	Perfect use of UML syntax	10
Software Implementation	1 or 2 user stories complete	3 user stories complete	4 user stories complete	Final product implements all core functionality, forming a viable product.	25
Continuous Integration	GitHub repository set up and in use	Everybody making commits, issue tracker being used	CI (eg. Travis-CI) integrated with badge on GitHub and able to auto build/run	Full use of GitHub including releases for this sprint	10
Test Driven Development	UATs being used for sprint 4, code coverage tool integrated on github	UATs in correct format, at least 40% coverage	Unit tests written properly, at least 60% coverage	At least 80% code coverage	15
Project Polish	Poor attempt at project so far given elapsed time.	Project is where we would expect it to be given the elapsed time.	Team has done work beyond what is expected of them given elapsed time.	Team on track for being in the top projects this year.	20
				Total	100