PROJECT NAME
Project Management Report

Client Name

Group number

Group members

CS2013 – CS3013: Sample Outline for a Management Report

1. The Project Planning Process

With regards to the planning of this project, our client laid out our requirements from the very beginning. At our first meeting as a group, each team member showcased their relevant skills and experience. When discussing the project description together, none of us were familiar with the concepts being asked of our client. After our initial meeting, we were still left quite confused but our client gave us great documentation breaking down the project into smaller concepts to tackle. This gave us a great starting point for creating teams.

We decided to split the team into 3 teams. 2 teams of two and 1 team of three.

1. Team One: Project Under Test
2. Team Two: Test runner and Configuration
3. Team Three: Docker and Kubernetes

For the first two weeks of the project, we set out and learned the technologies we would be using and researched new methods of development that we were not familiar with. Team members researched what technologies would be the best to implement the components. We had regular discussions with each other about our findings and how we thought it would be best to use what technologies and practices while keeping in mind the tight deadlines. We didn’t have much time to be spending on learning complex technologies we had no experience with at all. When we had decided the languages and technologies we would use, we began developing.

Even though we were split into three teams, each team still had to collaborate with the other to make sure when the 3 components came together there would be minimal tweaking to the code.

Every week we had our own meetings to discuss the progress we have made and what was next on the agenda for every team member whether it be documentation or development. We recorded our scrum videos in these timeslots. We also had meetings with our demonstrator to discuss any issues we had with our project or if we had any issues with our client. Everything went smoothly throughout the whole project and both our demonstrator and client were a great help to the team. They were quick to respond to any questions we had about topics relating to docker and Kubernetes and any other questions we had. We used Jira software to help manage the development cycles of our project. On Jira, we had the chance to report bugs and write up stories for upcoming sprints. This agile project management system gave us an insight into how large tech conglomerates would manage all their projects. Coming into the project you don’t expect to learn about these ways of managing. But it shows these tools are useful and very important in meeting our goals.

We met with our client weekly unless there were any issues with meeting times or our client had other matters to attend to. These meetings usually entailed us demonstrating our progress and getting feedback on what we could do better or what we could add to the project.

As a team of Second and Third years that did not know each other before this module, we worked amazingly well together. Every member had a pivotal role in all aspects of the project. Our communication was one of the main factors that made this a smooth process. Our management and control process over the course of this semester was very efficient when managing our time and resources in the development and documentation work needed for this module.

2. Project Goals and Objectives

Our client Ilya Birukov is a lead developer at Rapid7. Ilya gave us deliverables when we first met him. We were asked to create a Kubernetes load testing framework that will run requests against HTTP APIs, time their responses, and obtain these performance results.

The following components were needed for the success of the project:

Component	**Description**
Test Runner	A configuration file driven executor of tests, as configured.
Test Runner configuration	A file of a defined format that will tell the test runner what actions to perform.
Project under test	A simple HTTP server that will simulate a database.
Docker Containers	Will be used to package both the test runner and project under test.
Kubernetes	Will be used to take the docker container packages and run them.
Results UI	Graph resource utilization of the project under test using a JS or Python UI.

The team worked very well together to develop the load testing framework. The framework works how our client wants it too and he was very impressed with the implementation. One goal we could not meet was integrating it with Kubernetes. Containerisation and the orchestration of services is a difficult topic to grasp and fully learn while our team members have busy schedules with college and working outside of college.

That being said, we did learn a great deal about the topic and with a few more weeks the project would have been a complete success for us and the client.

As well as the goals we did meet, we did integrate a CI/CD system using AWS and Jenkins to automatically build our code and test it when it is pushed to GitHub. Our goal for implementing the CI/CD was to use it to package our code into docker images and run them in an isolated Kubernetes server using AWS EKS (Amazon’s Kubernetes Service).

Even though the project wasn’t deemed a complete success we have learned more from this project than we would have from picking a project we all would have been familiar with.

3. Project Scope

Outline whether there were any changes to the scope of the project . If so, describe and discuss how the changes were identified and handled and what impact they had.

3.1. Product Backlog

Provide the initial product backlog and the refined product backlog for each scrum sprint of the project.

Clearly describe and discuss how the changes in the refined product backlogs were: 1. Identified; 2. handled; 3. what impact they had.

4. Project Approach

Describe the scheduling, milestones and scrum sprints of the project and discuss whether the actual project was different to the project approach. If so, how was it different?

4.1. Scrum Sprints

Provide diagrams for the scrum sprints for the project and outline the rationale for the duration of the sprints.

For each scrum sprint provide the following: 1. objectives of the sprint; 2. the start and end dates of the sprint; 3. sprint planning summary; 4. sprint review summary; 5. sprint retrospective summary; 6. backlog refinement.

Comment ad discuss any relevant information in relation to the above.

5. Project Organisation

5.1. Staff

Provide the initial list of the team members (including their prior experience in projects and prior technical skills) and provide a list of the experience and technical skills acquired by each team member during the project.

Note: a table may be helpful for the above

5.2. Staff Chart

Provide the initial scrum staff charts outlining the following people for each scrum sprint in the project: Product owner; Scrum master; team members.

For each sprint staff chart outline changes (if there were any) to the work plan for each member of the team.

6. Risk Analysis

6.1. Risk Analysis

Outline and discuss whether any of the risks identified in the project plan materialised. If so, how were they dealt with?

Describe whether any new risks arose during the project. Outline them and discuss what impact they have and how they were controlled.

7. Project Controls

Factors	**Control Method**
Execution	We controlled the execution of our project by continually testing it and making sure there were no bugs or errors, and that the functionality of the project met the clients requirements specification. We ran and tested the execution of the project using various IDEs (Integrated Development Environments) along with external tools such as Postman.
Progress	We controlled the progress of our project by communicating with our teammates regularly in order to see if they are running into any difficulties that could slow down the progress. We aimed to complete each of our individual tasks before our team deadlines, so that in the event of a bug or other development issue that resulted in a delay, there was some additional time to finish things up.
Quality	We controlled the quality of our code by reviewing and giving feedback about any code that was contributed. Whenever any of us were uncertain about their piece of work, we would also ask for suggestions from our teammates before committing it to Github. We found this highly effective because multiple people are more likely to come up with improvements, find bugs, and offer great suggestions than just one person working alone on a task.
Deliverables	We controlled the deliverables by using teamwork and delegation. When we were assigned a new piece of work, we met up and planned it out as soon as possible. We split up the work into smaller chunks that could be worked on separately by everyone in the team. We delegated each subtask to the people that were the most comfortable with completing it. This was effective at reducing the challenges of learning new technologies.
Deadlines	We controlled the deadlines by aiming to start our work as soon as possible. We created soft internal deadlines and schedules when we were assigned tasks. This is so that we had some safe padding in the case of a disruption in the development. This also left us with time to review other people's work to give suggestions and feedback.
Communication	We control our communication using a voice and chat service called discord. We set up a channel and added all the group members to it at the beginning of the semester. We used it for our weekly scrum meetings because it was very convenient to do it there. We also used it to send in comments, files, updates, questions and feedback during the week while everyone was working on their task.

8. Communication

8.1. Client Communication

We met with the client every Friday at around lunchtime. This time was the most convenient for all of us as our timetables didn't clash as much on a Friday. We used zoom for the meetings because we were all familiar with it and it was easy to use. We sometimes gathered in person in the glassrooms and used a single device, and other times some of us were online and joined the meeting from home. During these meetings, we discussed our progress and asked for guidance. We performed a few demonstrations of the work we have done so far to make sure it was what the client wanted.

8.2. Project Team Meetings

We met with the team every week using discord just after the client meeting using zoom. It was the most convenient to do the two meetings back to back because we could discuss the client meeting afterwards. Friday was also a suitable date because it was the end of the week, and we could reflect on the work we did in the previous week, along with the plans we had for next week. During this meeting, we gave feedback, suggestions, analysed our progress and decided what we need to get done for next week, and set soft guidelines for the deadlines.

8.3. Demonstrator and Team Meetings

We met with our demonstrator every Monday at 6:50 pm. Here we discussed our progress and the demonstrator gave us some feedback. We also asked questions about things we were unsure about, and the demonstrator gave us some suggestions to help out. Our demonstrator was very helpful because she always reviewed any work we submitted to blackboard and talked about it in the meetings.