Rampage Organizational Document

Sprint 2
9/7/2023

Name	**Email Address**
Andrea Florence	andrea.florence180@topper.wku.edu
Tameka Ferguson	tameka.ferguson183@topper.wku.edu
Shikha Sawant	shikha.sawant059@topper.wku.edu
Aaron Beasley	aaron.beasley908@topper.wku.edu

CS 360
Fall 2023
Dr. Michael Galloway
Project Organization Documentation

Contents

List of Figures

1 Project Team's Organizational Approach

Sprint 1: When starting the project, the first thing we did was get all our resources needed for the project like getting an understanding which game engine to use when recreating the Rampage game. Most of our meetings were held during class time and sometimes through Discord. Even though most of our meeting times conflicted with other times like work, activities, etc., we are still working on trying to meet whenever possible. At the moment, we have mostly decided on meeting through Discord and at some point, we will also try to make a video to show that we are doing our part. Overall, we are still looking at ways to stay in full contact with each other.

We meet kind of periodically since we all have conflicting schedules with each other. So, we compromised that instead of just finding the right time, we could videotape ourselves doing each aspect of the project, so it doesn't look like we did it on a whim. Really, there is no straightforward way to make meetings easy for us without contradicting each other's times. But we do discuss what we are doing so that everyone is aware that we are still working on the project. Overall, like I said above, we are still in the process of figuring stuff out, but the actual project is running smoothly.

We didn't really decide on a leader for the project, but I think that Andrea was the main contributor to the entire project as she was the main help for everyone involved.

Sprint 2:

Sprint 3:

Sprint 4:

2 Schedule Organization

2.1 Gantt Chart v1:

The focus of Sprint 1 is to get started on our first technical and organizational documents as well as evaluations and presentations. Sprint 1's main focus falls under feasibility study and planning. As a team, we have analyzed the relevant factors of the project, such as the project scope, technical requirements, risk analysis, software process model, software product development, hardware/software requirements, schedule and timeline, costs/time costs, project visibility, deliverables, and team and client communications. With these factors we can get a better understanding of what the overall project requires and start planning ahead on what and how we are going to execute the requirements.

The Gantt chart was created by Tameka in Excel. It is updated a few times a week, and lists both group tasks and individual tasks. It is divided up by the four sprints, and color coded for each sprint as well. The four tasks we had listed for sprint 1 were the technical document, the organizational document, presentation, and evaluation. Each of those tasks were for everyone in the group, minus the creation of the slideshow for our presentation which Aaron did not assist with. Shikha created the slide layout and found the game style template, and Tameka and Andrea helped her add content to the slides. As for the documentation and evaluation – each person was assigned sections of documentation to do, and each person had to complete an individual evaluation of the team. Moving forward the entire group will be able to access and edit the Gantt chart as needed throughout the project. There were little to no individual tasks in this sprint of the project, whereas there will be many more individual tasks in the sprints to follow.

The Gantt Chart can be found in our Rampage zip file.

2.2 Gantt Chart v2:

Text goes here.

2.3 Gantt Chart v3:

Text goes here.

2.4 Final Gantt Chart:

Text goes here.

3 Progress Visibility

3.1 Sprint 1 Progress Visibility

The group uses a shared Discord server to communicate. Within the server, we have a channel specifically for task related subjects such as assigning them to people, tracking progress and completion of them, and asking for any help related to them when needed. This ensures that each member knows what tasks are assigned to them, and if they were to forget it makes it easy to go back into the channel and check what was assigned to them. When assigning tasks, the group makes of list of each thing that needs to be completed. We then allow each person to volunteer for tasks they feel confident in. If there are still tasks that need to be completed after that, we consider each member's strengths and delegate them from there. We also take into consideration how many tasks each person has at this point to ensure that no one member is getting overworked. If another person has to complete a task before someone can move on, the person waiting could tag them in the channel to give them reminders or ask for updates, and that pushes a notification straight to the other person. When that task is done, the same could be done in reverse to notify that person that they can begin their task. The group meets with the client once a week on Wednesdays at three-thirty to update him on the progress of the project.

3.2 Sprint 2 Progress Visibility

Text goes here.

3.3 Sprint 3 Progress Visibility

Text goes here.

3.4 Sprint 4 Progress Visibility

Text goes here.

4 Software Process Model

The software process model we are going to use throughout our project would be the Waterfall model. We chose this one because as we are going through one sprint at a time, before we move onto the next, we have to ensure that the previous requirements in the previous sprint are already fulfilled. We cannot move onto the next sprint unless we are done, so it is like the Waterfall model.

It increases our overall quality for the final deliverables because instead of having to try to do a rough draft of everything all at once and then going back to improve it, we can give our best for each step at a time. This is also helps us to focus on one step rather than all at once. The quality control steps of the Waterfall model would be feasibility study and planning (Sprint 1), requirements, system/ program design and modeling (Sprint 2), implementation (Sprint 3), and testing/maintenance (Sprint 4).

The feasibility study and planning was the analysis of the project's relevant factors, such as the overview, scope, technical requirements, production techniques, risk analysis, execution and development environments, schedule and timeline, costs, and project visibility and communication. The requirements define function of the system from the client's viewpoint. This establishes, system functionality, constraints, dependencies, goals, and the development process. The system/program design and modeling, describes the system from the software developer's viewpoint. The implementation is the coding of the software. The testing is to ensure that the code for the software works as intended.

5 Risk Management

5.1 Risk Identification

One risk is that the group could gain or lose a member at any time. Another is that the client could change their requirements and/or deadlines.There is also a chance that we have issues with meshing our code together.

5.2 Risk Planning

We will be keeping records of what each member of the group is supposed to do. If we lose a member, we will divvy out the tasks they had left to the remaining team members. If we gain one. we will each give some tasks to that member to balance out the workload between all members. If the client changes requirements we will update tasks as needed and update our time esitmation as well. If the deadlines change, we will consider task priorities and fullfill all completely necessary tasks first; tackling minor bugs and design with any time that may remain. If our code is not meshing, we will hold an emergency team meeting to work out the issues and rework things as needed.

5.3 Risk Monitoring

We will ensure that deadlines and requirements do not change at each client meeting. We will ask for prior notice of group membership changes, however we will be prepared for little to no notice of this risk coming to fruition. We will be testing code as a team in person, so will simply monitor code meshing issues as they arise in real time.