Team Supine - Project Management Review

We followed Scrum Methodology to manage our project. To prepare for this, we did online research to fully understand what this would entail, and built our plan around fundamental Scrum Principles of doing just enough planning to be prepared while still being flexible, postponing key decisions where needed to allow for new information and learning, and incorporating the customer into the dev team.

We met with our client twice before even starting development so we could get a good understanding of what he expected to get at the end of the project and what we can realistically produce. We spent a week after the first meeting brainstorming concepts and creating a proposal for him so we could get feedback on our idea and get the go ahead to start working on it. This meant that our client was a key part of the team from day 1, and had a key role in defining the product while also being kept aware of technical difficulties or time restraints that might make these ideas impossible. As an example of this, he originally wanted a game that could be played either with or without a child present as player 2. He was able to discuss this with our technical leads and understand that ultimately, the time taken to balance to different play modes would be better spent in making the main game as good as it can be.

From there, our Product Owner made a Roadmap giving us a rough outline of what we’ll be working on throughout the project and what each release should contain. This gave us enough information to know what we were working towards, while still being general enough that we didn’t have to pin down and be tied to specifics such as how goals would be achieved.

Using the Roadmap, the Scrum Master made a mock up of a Sprint Backlog to cover all of development. This document covered all the goals covered in the roadmap but broken down into a bit more detail. This sprint planning acted as a template for populating the actual Sprint Backlog, which was refined each release as the project evolved.

We chose to use Jira to manage our project, as we wanted to use industry standard software. We used the WatchTower add-on to allow us to organize tasks into a Kanban board, which is well suited to Scrum and is a concept we were all familiar with from experience with Trello.

We planned week-long sprints so that we touched base at least once a week, and were able to give updates to our client as regularly as possible.

At the start of each sprint, we held a sprint planning session to select tasks from the Backlog and move them to Sprint Backlog to be worked on. As we selected tasks, we assigned team members to them, and collectively agreed on a Definition of Done for that task, so everyone was on the same page about what that task entailed.

As we worked on tasks, they were moved to the Doing column. Once completed, they were moved to the ‘In Review’ column, indicating that they needed to be reviewed by another team member. This was mainly for code-based tasks, as it allows everyone to have a chance to read through others work and understand it, and check that it’ working as outlined in the Definition of Done. This also allowed the Codebase Manager to look through and ensure the practices we agreed on relating to commenting and naming were being followed.

Once peer-reviewed, tasks were moved to Sprint-Done column, so we could see clearly what was finished with for the current sprint.

At the end of the sprint, we held a sprint review meeting to discuss the work we’d completed and any issues we found. This helped ensure everyone was caught up on the stage of the product, and gave the Product Owner enough information to write up a weekly roundup to be sent to our client, keeping him in the loop. We then moved all tasks in Sprint Done to Completed Tasks, and held our Sprint Planning meeting to prepare for the next sprint.

Halfway through the project, we started holding life-coding sessions to cope with working remotely. We all worked on our tasks as designated times while in a voice call together so we could discuss our work and any issues. This made up for not being in the same place, and helped motivate team members to work on tasks.

Our project management helped keep the work rate steady, and having designated days for meetings and updates ensured we didn’t fall behind with contact with one another and our client, especially in the latter part of the project when we had to make the shift to remote working.

Generally, the project was well managed, but there are some areas we would aim to improve upon if we were to redo this project. The biggest issue was breaking down tasks into cards for areas the Scrum Master didn’t have a good understanding of, such as web development. These areas were generally left as quite vague tasks (such as ‘Make Player 2 Drop Items’) which weren’t small or specific enough to be done in a day, or even in a single sprint. When these tasks were added to the Project Backlog at the task, the plan was to narrow them down and add specifics during backlog refinement, but this was often neglected. As a consequence, it was harder for the team member assigned to these tasks to keep track of what work needed to be done. With hindsight, this issue could have been avoided had we scheduled meetings between the technical leads and the scrum master to properly discuss what each task would entail and update cards accordingly.

Supporting Evidence

This section contains all the key agile artifacts produced to demonstrate our teams engagement with the development progress.

Product Roadmap

* This roadmap was produced at the start of the project by our Product Owner, Zack, following discussions with our client and the team about what we can deliver.

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **20th March 2020** | **10th April 2020** | **1st May 2020** |
| Name | RELEASE 1 - CORE MECHANICS AND CONNECTIVITY | RELEASE 2 - GAMEPLAY | RELEASE 3 - POLISH |
| Goals | Get the core mechanics of the gameplay implemented and working properly. Also, have the connectivity to the server and back to the tablet device. A couple of rooms, bare bones. | More complex aspects of the project, combat between enemies and player. Weapon and item pick ups. Goal here is to get the majority of the functions and complexity in so the game is mechanically finished. Variety of rooms can be generated procedurally. | Final release of the game, No major bugs that affect gameplay all elements of the gameplay and mechanics are fleshed out properly and finished. Polished with graphics and lighting as well as audio etc. UI |
| Features | * Player movement * Simple camera movement * Server connectivity from tablet to PC * Couple rooms to test gameplay * Simple enemy AI * Basic version of tablet game for testing purposes. * Mini game made, not implemented | * Final camera movement * Procedurally generated levels * Variants of AI for enemies to match which enemy they are * Different games for the tablet user to complete and different ways they interact with the level * Combat, user can use different weapons they pick up to engage the enemy * Pickups | * Complete game loop * Models all rigged and animated * Audio, music for menu and game as well as sounds for hits etc * Lighting, baked as well as realtime. * UI * All of the level variation we want. * All types of games for the tablet user to play. * Menus |

Product Backlog

* The backlog is populated by the Scrum Master, Josie, using the roadmap as a template. It is dynamic, and detail is added/removed to cards throughout development by the whole team as part of backlog refinement.

// Insert screencaps of the jira, and a full list of tasks when we’ve completed the project

Sprints

* Sprints are planned every week. The Scrum Master takes the lead in running a Sprint Planning session, in which we go through the backlog as a team and select tasks to move to Sprint backlog and work on in the coming week. We then go through each task, adding detail as needed, and agreeing on a Definition of Done so everyone is on the same page as to what that task entails. We then assign team members to the tasks.
* To ensure everyone kept up with our Scrum plan, our Scrum Master produced an overview document at the start of the project outlining all the different sprint events to be covered and how we’ll manage them. (Shown Below)

Fun Scrum Overview

Project Management will be done in Trello [here](https://trello.com/b/NYCEYjfy/prco204-supine).

Each list has a card which gives an overview of what that scrum item is and how to use it, so we can refer to them as needed throughout the project to refresh our memories.

Each sprint will have these events:

1. Sprint Planning – as soon as one sprint ends, we will hold a Planning session to prepare for the next sprint. All team members will be present, and we will work together to pick tasks from the backlog, define the definition of done, and add details where needed, and and add them to the Sprint Backlog. From the selected items we will determine a sprint goal. Throughout the sprint, items will be selected from the Sprint backlog to be worked on.
2. Daily Scrum – in work environments, all team members will get together each day to discuss what they achieved the day before, what they plan to do today, and what might prevent them from achieving it. This should be quick, just a chance for everyone to reflect on their progress and make sure everyone is aware of the teams work. This can be done in Slack to show we’re touching base and be used for our documentation to show our agile workflow is being followed.
3. Backlog Refinement – During each sprint we should look back on the backlog and update any items that need refinement or are no longer necessary.
4. Sprint Review – At the end of a sprint, we will get together to discuss what we’ve achieved. This would be a good meeting to have with Luke, as we can demo what progress has been made, get feedback on completed work, and update the roadmap and backlog with any new requirements he suggests.

Proposed Weekly plan

If we have week long sprints, I suggest we organise them Friday to Friday.

As Luke is available for meetings between 4 and 5, we can spend our 4 hour morning session going through our Backlog Refinement, collating work completed to demo for Luke, and planning our next sprint. We can then have a few hours of to chill out, and head back in for 4pm to meet with Luke and hold our Sprint Review.

* When we went into lockdown, we moved our weekly meetings to online voice calls. We held these on Friday/Saturday evenings to best fit around everyones new schedules. While it was difficult not to meet face-to-face, screen sharing capabilities allowed us to explain any issues we had clearly and meetings continued to be effective.

Release Plan

// need to produce a release plan, using roadmap as guideline

Summary of group meetings

* We document our meetings on Github classrooms. The Scrum Master makes a post before each meeting with bullet points on what to cover, and then adds conclusions of the meeting when the meeting is over.

// add documentation of HCI feedback

//Hold an informal interim review and document it here

Official Meetings with client

* At the start of the module, the Product Owner organized fortnightly meetings with our client to be held on campus with all team members. We held 3 meetings this way, documented below. We also maintained constant contact via Slack, so we could reach out at any point and get feedback as issues arose.

// add minutes of official meetings/recordings

* When we went into lockdown, communication with the client was understandably strained. Our implantation of Slack proved useful, as we were all able to reach out to Luke, and our Product Owner could send weekly updates to keep the client up to date.
* As time went on, our client stopped responding to weekly updates. There was a period of 2 weeks where we heard nothing, despite repeatedly reaching out via email and Slack. To solve this, our Product Owner secured our clients phone number, which the Client said was more likely to check on a regular basis. Going forward, we maintained use of Slack for long-form updates and information, but sent text messages for important information such as confirmation of meetings or any issues in development.

// documentation of this.

List of issues encountered and solutions found

// everyone should write up any of the issues they found in the given table

|  |  |  |
| --- | --- | --- |
| Team Member | Issue Found | Solution |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |