# Presentation Transcript

**Group 1 – Alice Villar, Hendrick Van Rooven, Gennaro Coppola, Sharon Wong.**

**SEPM Artefacts & Prototype:**

* GITHUB REPOSITORY:

<https://github.com/SEPM-2022/CleverBirds>

* CONFLUENCE:

Confluence PDF on Github Readme file – Section 11.

* VIDEO PRESENTATION:

<https://drive.google.com/file/d/1ey6QWLDuiZW_9KxYp4D29YSA3fxOb3vb/view>

# 

|  |
| --- |
| Introduction |

# Presentation with music.

# Scrum Team is introduced.

|  |
| --- |
| PART 1 |

Hi everyone,

Before we begin, please allow me to introduce myself. My name is Alice Villar and I’m the project manager of a team of four members who developed the application “Clever Birds”.

In this presentation I will go through the following topics: Project Overview, team governance and processes, project conception and life cycle, application demo, planning, architecture and design, build, testing, training and support and User documentation.

Throughout the development of this project, we used Confluence and Jira together for planning and coordinating our work. In this presentation I will show you how we used Confluence to develop our product documentation, track meeting minutes, draw process flow diagrams, and create technical architecture documents. We will also show you how we used Jira to assign tasks and subtasks to our whole team, as well as to manage the workflow and track the progress.

* Project Overview

This presentation is focused on the project management process of building our application. Here are the five stages of project management: conception and initiation, definition, and planning, launch or execution, performance and control, and project close. Here in this diagram, you can see some of the main artifacts that will be part of this presentation.

|  |
| --- |
| PART 2 – Team Governance & Processes |

Welcome to our Confluence documentation space. Our first page presents our team governance and processes. First, we explain the reasons why we chose Scrum. In short, we chose Scrum for its simplicity and high performance. Scrum encourages teamwork, is easy to use, saves time, embraces change and isextremely effective and efficient.

Here we document our Scrum events, which helped us to continuously optimize performance.This diagram presents the Scrum lifecycle, a number of consecutive steps and iterative stages that should be performed during the realization of any Scrum project.

After defining the Scope of the project (which I’ll show you later in our project charter), we did our Scrum Product Backlog. Simply put, it’sa list of all things that needs to be done within the project. Then we came to the design phase. In this phase, the product backlog should be up-to-date and refined to provide clarity. Finally, we have the Sprint planning meeting, when the team meets to determine which backlog items will be handled in the next sprint. It’s the Sprint backlog.

The sprint automation, also known as in-sprint test automation, which portrays an ideal development scenario where the entire testing process—from creation and implementation to execution and reporting—happens in one sprint.

TheSprint execution is the work the Scrum team performs during each sprint to meet the sprint goal. It includes task planning, managing the tasks and attending daily stand-ups. Then we have the sprint review meeting at the end of a sprint, in which the Scrum team shows what was accomplished during this period. Finally, the sprint retrospective, which is an opportunity for the Scrum Team to inspect itself and create a plan for improvements for the next Sprint.

With Confluence we could document our five scrum ceremonies so we can all be on the same page. Each member of the team can come here to know what is expected from them in each meeting.

Our team had weekly meetings as an additional opportunity to discuss progress made. These meetings are documented here, on “meeting notes”. Here you have access to our RACI chart, a visual representation of the functional role played by each person on a project team: project manager, technical writer, developer and product owner.

Here we documented our Definition of Done, which is crucial for ensuring quality. The purpose is to provide transparency about what it means to be “done”.

|  |
| --- |
| PART 2 – Performance & Control |

Let us now talk Performance and Control, where we measure of how efficiently we worked together. Here we present ourBudget control charts. The first, called Budget analysis is comparing how much time we estimated to spend in each Epic with the actual time we spent working on it. The second graph shows the amount of work of each team member.

Here is the last topic of the Team Governance and Processes: Gitflow workflow, atrunk-based workflows which are now considered best practices for modern continuous software development. Confluence documentation helped a lot. Here we defined step by step how to work with Gitflow workflow. We recorded a video to help us on the process and posted it here.

Managing a project with multiple branches, commits and developers working on one set of code at once can get messy and complicated. To better visualize and manage our repository we used SourceTree, a graphical user interface that simplifies how we you interact with Git repositories. The key benefit is the visualization of the Git process.

This section displays the full history of the current branch. These colored lines are the features. When they close back you know that the feature was properly finished. When it’s open like this it’s either because the developer is still working on it or because it wasn’t finished correctly. We had few mistakes done in our tree which served as a valuable learning experience. Gitflow ensures that parallel code changes are completed without many hiccups. We experienced a huge reduction in merge conflicts thanks to GitFlow.

|  |
| --- |
| PART 3 - Project Conception & Life Cycle |

In this page we present our project charter, the requirements gathering phase and the project close. This section contains: Functional requirements, Non-functional requirements, and High priority requirements.  From the agreed ten requirements, we have chosen five as a high priority, due to their importance to the overall product functionality and design. This presentation is focused on the high priority requirements.

This is our project charter, where we define the project description, goals, high priority requirements, deliverables, game description, methodology and schedule. Here in the Project close we have the Project Post-Mortem Retrospective.

|  |
| --- |
| PART 4 – Application DEMO |

Here are our five high priority requirements for demonstration are:

**1) Player persona**: In order to register, you have to provide your name, username, password, gender and avatar. The application allows users to choose one of the following three avatars: Daisy, Birdy, and Alfredo.

**2) User Profile**: After registration, users can see their personal profile. For registration, you have to provide you name, password, user name, email, gender and avatar.

**3) Cancelation of subscription**:  On the page "Manage Account", users can delete their account, which will erase their data from our database.

**4) Safety**: The chatbot "Tweety" teaches parents how to protect their children by discussing video games' health problems and how to solve them.

**5)  Simplicity**: Clever Birds is very simple. Users can see a “play now” button on the top right on all screens. By pressing this button, the user starts playing.

|  |
| --- |
| PART 5 – Planning |

On the page planning, first we have our product vision which presentsthe reasons for creating the product. Then we have a short summary of our sprint plans, presenting the duration, the objectives and who will participate.

Here is our project estimating, which we developed using the Planning Poker technique. In our project estimating, 1 Story Point = 4 hours. In our Gantt Chart, 1 day corresponds to 4 hours of work. Thus, we have 28 hours per week. Each Sprint has two weeks and therefore 56 hours of work. Each Sprint has 14 Story Points, so we have 56 hours.

Here we discuss and present our risk management and quality control. Agile methodology has an iterative approach that enables continuous attention to risks. Quality management focus on improving, developing, and testing processes with the aim of preventing defects. Our strategy was to prevent the defects by doing analysis of the results, and improvement actions. We also had defect detection in the completed product, which is better known as Quality Control.

Finally, we present our Gantt Chart, a graphical representation of activity against time, illustrating a project plan.

|  |
| --- |
| PART 5 - Architecture & Design |

Here on Architecture & Design page we have the Software Design Document (SDD), a comprehensive software design model consisting of four distinct but interrelated activities: interface design, database design, architectural design and procedural design.

Here You can see our interface design. We have 10 pages. Here is our Database design. This is our entity-relationship diagram, automatically generated with SQL-Server Management Studio. Then there is our architectural design table and the procedural design, showing our activity diagram.

|  |
| --- |
| PART 5 - Build |

On the build page you have access to our project structure.Our project follows the Model-View-Controller (MVC) pattern, an architectural pattern consisting of three parts: Model, View, Controller. This "separation of concerns" provides for a better division of labor and improved maintenance. Getting Started Page is where we explain how to Run application and how to create Ngrok to expose local server ports to the Internet.

|  |
| --- |
| PART 6 - Testing |

We have three types of tests: Unit Tests, Integrated Tests, User acceptance testing (UAT). In our testing framework we present the test scope, principles, objective and types, defect management, documentation and tools.

Jira Test Management tool helped us test, plan, track, and release our software. This is a demonstration of our Release test Plan. Here in the Feature Testing. We did Unit and Integration tests, which are documented on Confluence. The User Acceptance Test was done by Sharon Wong (our customer service) and Group 1. Finally, we have our Release Acceptance Testing in the end stage of the testing process, which we did with Xray (Jira addon).

|  |
| --- |
| PART 7 - Training & Support / User documentation |

To keep our team sharp and up to date, we created a number of pages for training and support. Finally, here is our user documentation. In our application we have two pages Page “about us” and “privacy policy”. Plus, we have the Readme file on Github.

* Small video presenting github readme file.

|  |
| --- |
| PART 8 - JIRA |

Here is our JIRA workspace. This is our backlog, a set of issues that await planning and implementation. The backlog is a space for defining and prioritizing work our team will take on. In Jira Software, issues typically represent things like features, user requirements, and software bugs.

I will now show you two different issues, a bug and a story.

* Using a Jira bug reporting tool helps to record and track the status of bugs. We can add a bug description, assign them to the right members to fix, and assign severity levels, all while having a single overview of everything in the backlog.
* Here is a story card for the registration form. In the description we included the behavior description, details and software security requirements. Here are the subtaskes, linked issues and acceptance criteria. On this side you can see the details of the task. It has two story points. As I mentioned earlier in this presentation, in our project estimating one story point corresponds to four hours. In our Gantt chart, one day corresponds to four hours of work. Thus, when we say that the registration form has two story points it means eight hours of work. I will now bring the Gantt chart so we can see this information there too. The registration form is in the second sprint. It is assigned to Gennaro and corresponds to eight hours of work.

|  |
| --- |
| PART 9 - Closing |

## This brings me to the end of my presentation. I have shown you how we used JIRA to plan and track the project work. Adding Confluence gave us the power to organize all the ideas, content and files. That ladies and gentleman was what I wanted to share with you.