Project Plan

Quantum Chess Tournament Online Platform

Fontys Engineering Quantum

Date	:	20-02-2024
Version	:	1.0
State	:	State
Author	:	Author

Version history

Version	Date	Author(s)	Changes	State
0.1	20-02-2024	Whole team	Started the document	Basic start
	26-02-2024	Esther Wolfs	Answer chapter 2 and 3 with more details	
	26-02-2024	Lucas Jacobs	Answer chapter 1 with more details	
	29-01-2024	Maike Meek	Review the chapters	
	29-01-2024	Genelle Geerman	Adding to the testing strategy	
	01-03-2024	Maike Meek	Process feedback	

Distribution

Version	Date	Receivers
0.1	29-02-2024	Gerard Elbers
1.0		

Contents

1. Project assignment

1.1 Context

The Quantum Talent and Learning Centers (TLC) Eindhoven is focused on advancing education in quantum tech, information, and sensing. The goal of Quantum TLC is to promote and ensure education at various levels, including WO, HBO, MBO, secondary, and primary schools, as well as those seeking a career change.

To help with their goal, we have been asked to make a software solution to make an online platform to host Quantum Chess Tournaments.

1.2 Goal of the project

Quantum Talent and Learning Centers (TLC) Eindhoven has created a game called Quantum Chess, which is a variant of chess with dynamic effects of quantum mechanics. Right now, they can only play a game on one computer, making it so that users need to be physically together with one computer. The task that the software developers need to solve is to offer a software solution where users can use an online platform where they can join tournament-hosted games held by Institutes. This system is used for engaging and collaborative competition that spans all four Quantum Talent and Learning Centers (TLCs) in the Netherlands. A user can log in to the system, sign up for a tournament and play online hosted games. The focus will be on a user-friendly design, participants can sign up and smoothly take part. This end goal is to have a working solution with a great approach to future working software developers for this project.

1.3 Scope and preconditions

Inside scope:	Outside scope:
1 Solution: Online Tournament platform	1 Mobile- and PC application,
2 Hand-over document	2 Embed multiple games on the website
3 Documentation (Technical, project plan)	3
4 Embed the quantum chess game on the website	

Users at the same time around fifty people.

Quantum TLC set some preconditions; these are the following:

- Solution needs multi-language support.
- The solution needs to be GDPR compliant (important to manage players' data).
- To enable a smooth project handover, the solution needs to be well documented, and the project code should be of high quality (e.g., readable, maintainable, extensible, etc.).> Additionally, solution must work as it will be used for a real event, so hosting should be stable.
- Regarding the security: There is no cheating allowed in the game.
- Needs to be future proof, the games need to be hosted every year/semester.
- Required roles: Player, Admin

- It needs to be able to oversee many concurrent players. This is discussed with the product owner, and it states that the solution will be aiming at around fifty users a at the same time.

1.4 Strategy

In this project, we work with an agile approach. Each sprint will consist of 3 weeks. The project will have a design phase (Sprint 0), where the team will get to know the product, and product owner, and make the documentation to get a better insight into understanding the project assignment.
After the design phase, each sprint consists of four phases:

- Sprint planning (week 1): Create a plan for the upcoming three weeks, we select which user stories we will be working on during the sprint.
- Analyze (week 1): Doing research, thinking about what research methods to use to answer the research questions. The main goal is to have a basic understanding to start implementing the solution.
- Realize (week 2): Start implementing the solution.
- Evaluate (week 3): Finishing the work, at the end of the sprint prepare a demonstration for the client and a delivery.

After each evaluation of the sprint, the received feedback and new insights will then be applied to the next coming sprints. Furthermore, the meetings with the client will be every three weeks (sprint review/planning for the next sprint), with rotating on-site and online.

1.5 Research questions and methodology

Research Methods for Project Investigations

Main: “How to develop a user-friendly online tournament platform for Quantum Chess that meets the needs of the admins and users, while keeping in mind the legal regulations and a stable hosting environment?”

Sub-questions:

1. What are the existing online tournament platforms, how do these platforms operate?
    a. What do these platforms offer, like what technology, approach, and UI/UX?
    b. What are the main strengths and weaknesses of these platforms?
2. What are the specific needs and preferences of users interested in participating in a quantum chess tournament?
    a. Are there any specific prioritizations to the functionality of these people?
3. How can it be validated that the solution that will be used by Quantum TLC Eindhoven ensures its readiness?
    a. Are there any criteria that need to be met to say the system's satisfaction, keeping in mind any standard requirements?
    b. What testing and quality measurements can be considered to ensure the quality of our product?
4. What is the most suitable system for a stable application?
    a. What building framework for front-end best fits our project?

        i. How scalable does the project need to be, and does this work with the building framework?
        ii. How big is the community and support for the framework?
    b. What building framework for back-end best fits our project?
    c. What database framework is a satisfactory solution for the tournament system?
    d. Is there a needed architecture that enhances performance?

5. What are the technical considerations and challenges when it comes to hosting online tournaments for a game like Quantum Chess?
    a. What technologies are commonly used when it comes to hosting games in a one versus one format?
        i. Which platforms are dominating this industry?
        ii. Is there a commonly used architecture for implementing hosting technology?
    b. What are the main scalability and performance requirements when it comes to hosting a game?
6. What are the legal and regulatory requirements for hosting online gaming tournaments, with the consideration of GDPR?
    a. What are the main data protection rules/requirements?
    b. How do we apply the measurements of GDPR in our application?
7. What are the best practices when it comes to security of the website and the zero-tolerance cheat policy in online gaming tournaments?
    a. Which security measurements are considered when it comes to online gaming platforms?
    b. How can cheating be detected and prevented effectively?
8. What are the main user interfaces and experiences expected when it comes to online hosted tournaments?
    a. What design elements in an online gaming platform contribute to a user-friendly application?
    b. How can we improve the experience when it comes to using the application on different devices?
9. How can we make sure that hosting and infrastructure are performing stable enough during a tournament?
    a. What hosting solutions can oversee fifty plus users at a concurrent time?
    b. How can we limit the performance and potential bottlenecks?

Research Methods

Let us now go over each sub-question and add the needed research methods using the DOT framework to answer the questions.

1. What are the existing online tournament platforms, how do these platforms operate?
    a. **Best good and bad practices:** Look for platforms where the existing system is operating as promised. Look for what other systems have done and check if it can integrate inside the solution.
    b. **Community Research:** Look at what the community of certain online platforms has to say about them.

   c. **Literature Study:** Look for any information regarding online tournament platforms and why they applied certain strategies like why they have chosen certain elements, colors, and hosting systems.

2. What are the specific needs and preferences of users interested in participating in a quantum chess tournament?
   a. **User Requirements Exploration:** Gaining comprehensive insights into user needs and preferences for the proposed solution.
   b. **Focus Group:** Collecting diverse perspectives and insights on the issue at hand.
   c. **Interviews:** Conducting interviews with stakeholders to delve deeper into their opinions and requirements.
   d. **Stakeholder Analysis**: Identifying and addressing the needs of relevant stakeholders.
   e. **Prototyping:** Developing prototypes to materialize conceptual ideas.
   f. **Brainstorming:** Generating innovative ideas through collaborative brainstorming sessions. Also, in the HackIT (week 1), having sessions with each other to produce ideas.
3. How can it be validated that the solution that will be used by Quantum TLC Eindhoven ensures its readiness?
   a. **Product Review:** Giving a demo and reviews to the stakeholders every sprint, to get continuous feedback from them. This gives a good insight into what requirements need to be met.
   b. **Unit test:** verify that the logic inside the code is working as supposed. Make sure the code is up to quality and prevents unwanted behavior.
   c. **Code review:** When a person is done with his functionality, another person in the group will review the code to find bugs, coding standards, and improvements.
   d. **System tests**: having a document with different test flows of functionalities to assess a case with the expected outcome.
   e. **Peer Review:** Collaboratively reviewing code using GIT to detect and rectify minor errors that may slip through unnoticed.
4. What is the most suitable system for a stable application?
   a. **Literature study:** Getting knowledge of the different systems that are available.
   b. **Community research:** look for communities to get their perspective on different kind of software.
   c. **Comparison Chart:** Creating a comparison chart to evaluate various plugins based on factors such as cost and effort.
5. What are the technical considerations and challenges when it comes to hosting online tournaments for a game like Quantum Chess?
   a. **Competitive Analysis:** Look at what the main dominant applications are when it comes to hosting an application.
   b. **Literature study:** Getting to know what architecture can be used when hosting tournaments, making a trade-off about which one will be preferred in this case.
   c. **Design pattern research:** Look for certain patterns that can be applied to have a good architectural structure.
   d. **Comparison Chart:** Creating a comparison chart to evaluate various plugins based on factors such as cost and effort.

6. What are the legal and regulatory requirements for hosting online gaming tournaments, with the consideration of GDPR?
    a. **Literature study:** Looking for the regulations of the GDPR. What is it? How to apply it?
    b. **Community research:** Get to know the ways other people have implemented these GDPR regulations.
    c. **Available product analysis:** Looking for existing products that apply regulations and look for solutions that can be applied for this project as well.
    d. Problem Analysis: Thoroughly understanding the problem landscape. So why GDPR?
7. What are the best practices when it comes to security of the website and the zero-tolerance cheat policy in online gaming tournaments?
    a. **Literature study:** getting to know what the best options are to securing this application, what the best software solutions can be against cheating in a game.
    b. **Best good and bad practices:** Looking at secure system solutions.
    c. **Comparison Cha**rt: Creating a comparison chart to evaluate various plugins based on factors such as cost and effort.
8. What are the main user interfaces and experiences expected when it comes to online hosted tournaments?
    a. **Product review**: Throughout the project, having enough meetings with each other, discussing the looks with the stakeholders.
    b. **Usability testing:** Other users can investigate the product and give feedback. With this test, we can assess certain design choices and fix issues before the actual product goes live.
    c. **Community research:** look at what people find a nice UI/UX regarding a tournament system.
    d. **Literature study:** Look at the best design principles for creating a clear UI/UX.
    e. **Problem Analysis:** Thoroughly understanding the problem landscape.
9. How can we make sure that hosting and infrastructure are performing stable enough during a tournament?
    a. **Code review:** Look at if the structure of the hosting is set up correctly. The system is separated with the correct specifications.
    b. **Literature study:** Looking at the best solutions to performing and having a stable application.
    c. **Model validation (ML):** check if the system can oversee the expected traffic.
    d. **Tinkering:** Engaging in trial-and-error coding to identify optimal solutions.

1.6 End products

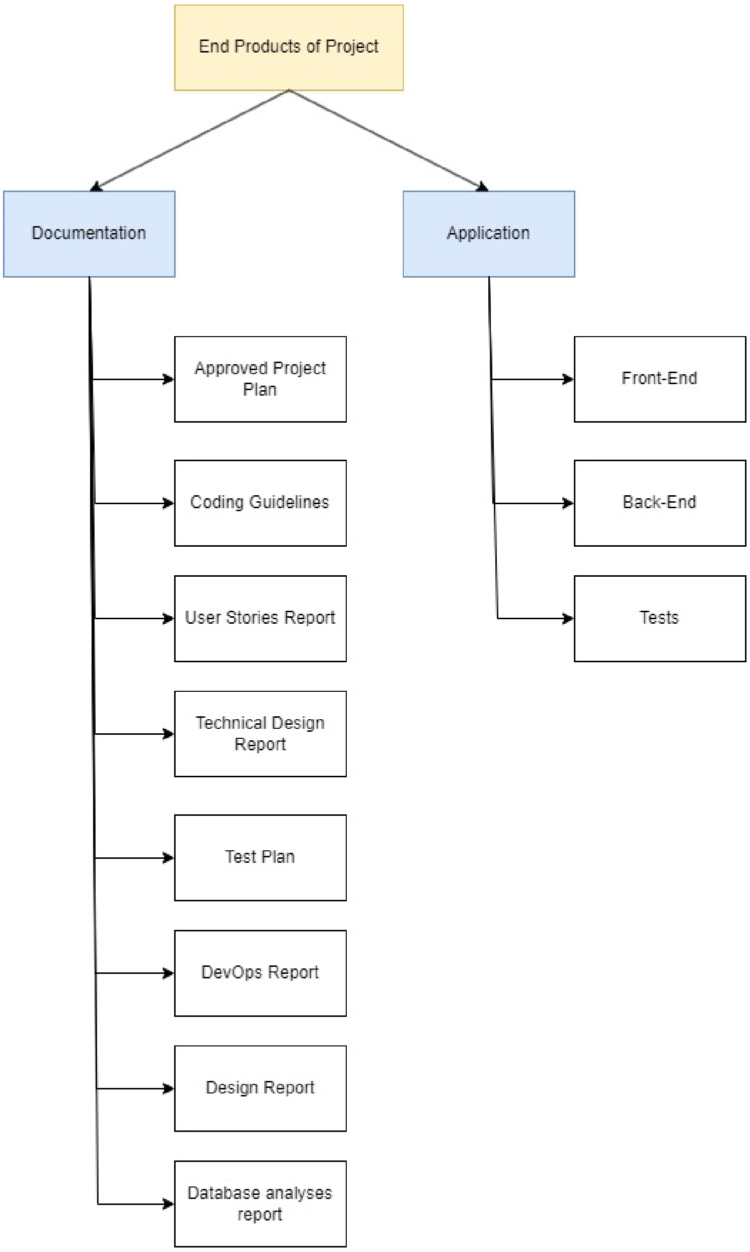


Figure 1: End products overview

Products description

Documents:

- **Project Plan:** Breaking down the project, what the assignment is about, how we are going to approach the project, contact information, risks and budget, and the testing strategies.
- **Coding Guidelines:** The specific rules every developer of the project needs to keep in mind when coding.
- **User Stories Report:** All the requirements for the application in the format of user stories. Ordered using the MoSCoW prioritization format.

- **Technical Design Report:** Specifying all the technical reasoning, security measures, diagrams (flow charts, sequence diagrams, etc.).
- Test plan: the functionalities that need to be assessed inside the application.
- **DevOps Report:** How did we manage an environment with CI/CD, what are the reasonings behind those decisions?
- **Design Report:** All the decisions of the design regarding the application, a page overview diagram, and wireframes.
- **Database analysis:** Why we chose a certain database, the structure, and meaning of each table.

Application:

- **Front-end:** all the code of the front-end application.
- **Back-end:** the code of the back end of the application.
- **Tests:** all kinds of testing to have a well-working application.

2. Project organisation

2.1 Stakeholders and team members

Name	Abbreviation	Role and functions	Availability
Lucas Jacobs l.jacobs@student.fontys.nl	*LJAC*	*Developer*	*Monday, Wednesday, and Thursday at TQ5.*
Genelle Geerman g.geerman@student.fontys.nl	*GGEE*	*Scrum Master/ Developer*	*Monday, Wednesday, and Thursday at TQ5.*
Maike Meek m.meek@student.fontys.nl	*MMEE*	*Contact person/ Developer*	*Monday, Wednesday, and Thursday at TQ5.*
Gabriela Hristova g.hristova@student.fontys.nl	*GHRI*	*Developer*	*Monday, Wednesday, and Thursday at TQ5.*
Esther Wolfs esther.wolfs@student.fontys.nl	*EWOL*	*Developer*	*Monday, Wednesday, and Thursday at TQ5.*
Mohammad-Amin m.moradi@fontys.nl 0617478678	*MMOR*	*Product owner*	
Gerard Elbers g.elbers@fontys.nl	*GELB*	*Semester coach*	*Thursdays at TQ5*
Bartosz Paszkowski b.paszkowski@fontys.nl	*BPAS*	*Technical Expert*	*Monday morning, Wednesday at TQ5*
Felipe Ebert f.ebert@fontys.nl	*FEBE*	*Technical Expert*	*Monday at TQ5*
Basten van der Vorm g.vandervorm@student.fontys.nl 0637421755	*BVOR*	*EduQuation Team Member*	
Bas Hankel b.hankel@student.fontys.nl	*BHAN*	*EduQuation Team Member*	

Bram Spijkerman b.spijkerman@student.fontys.n	*BSPI*	*EduQuation Team Member*	
Guus Hendriks guus.hendriks@student.fontys.nl	*GHEN*	*EduQuation Team Member*	
Imke Houben i.houben@student.fontys.nl	*IHOU*	*EduQuation Team Member*	

2.2 Communication

On Monday, Wednesday, and Thursday the developers will have a stand-up meeting at TQ5, where we will discuss what we are working on and the progress we have made.

NAME	PEOPLE PRESENT	DAY	FREQUENCY	GOAL
STAND UP	Lucas, Genelle, Maike, Gabriela, Esther, (Gerard)	Monday, Thursday	Twice every week	Know people's tasks and updates of the week
SPRINT PLANNING	Lucas, Genelle, Maike, Gabriela, Esther, (Gerard)	Thursday	Once every three weeks	Set new goals for the upcoming sprint
SPRINT REVIEW	Lucas, Genelle, Maike, Gabriela, Esther, (Gerard)	Thursday	Once every three weeks	Reflect on previous sprint
PROGRESS MEETING	Lucas, Genelle, Maike, Gabriela, Esther, Gerard, Mohammed, EduQuation team	Thursday	Once every two weeks (online/ onsite every other meeting)	Show the progress (sprint review), get feedback, and discuss upcoming goals (sprint planning) for the upcoming two weeks.

Every three weeks there will be a sprint review with the developers, the product owner, the quantum chess team members, and the semester coach. The goal of these meetings is to show the progress made during the sprint and discuss the planning for the next sprint. These meetings rotate between being online and on site, at either TQ5 or TU/e. For these meetings we create meeting minutes that we save in a separate folder so we can have an overview of decisions made and action points for both sides as well as knowing who was present. These are in the same format as shown below in the example:

"""
Minutes of progress meeting 22/02/2024
Present:
Maike Meek, Gabriela Hristova, Lucas Jacobs, Genelle Geerman, Esther Wolfs
Mohammed-Amin Moradi, Sjoerd Wijgerse
Gerard Elbers

AP = Action point
D = Decision
Next Meeting: On-site (TU/e, Nexus Building), 7th March 9am

//AP (Maike, 22/02/2024): Send contact information and PowerPoint presentation

//D (22/02/2024): Guest spectators, you can view the games without having to be logged in
"""

For questions with the stakeholders, we will use email.

To show the progress of the project to the teachers, we will ask for feedback at least once every two weeks.

3. Activities and time plan

3.1 Phases of the project

This project consists of six sprints of 3 weeks, 18 weeks in total. Each sprint starts with a sprint meeting with the developers, where we will update the backlog and problem analysis. At the end of the sprint, the developers will meet with the stakeholders to show the sprint's progress and discuss the next sprint's planning by selecting new user stories to work on. This way we will know what user stories we will be working on during the next sprint. At the end of the sprint the developers will have a retrospective, to reflect on the sprint.

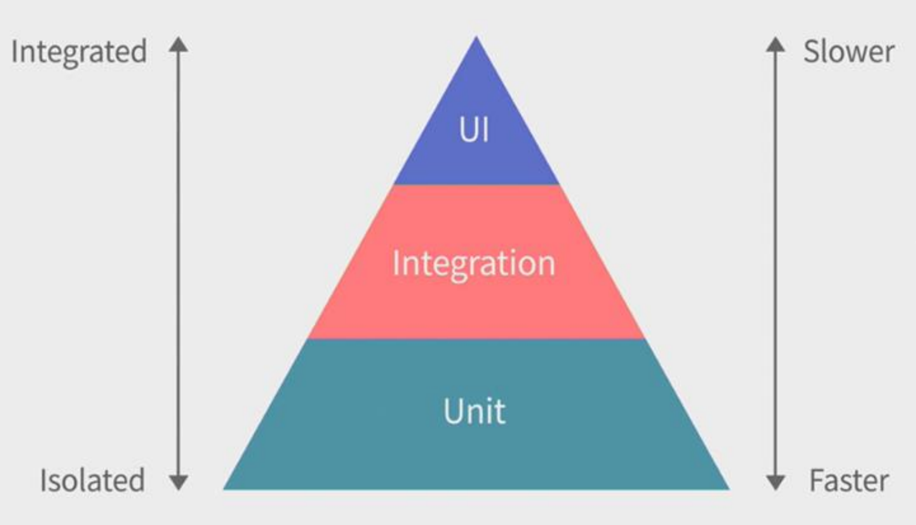
3.2 Time plan and milestones

Phasing	Effort	Start date	Finish date
1 Sprint 0 – Work on project plan, make backlog, make design decisions	3 weeks	19-02-2024	08-03-2024
2 Sprint 1 – Work on backlog	3 weeks	11-03-2024	29-03-2024
3 Sprint 2 – Work on backlog	3 weeks	02-04-2024	26-04-2024
4 Sprint 3 – Work on backlog	3 weeks	06-05-2024	24-05-2024
5 Sprint 4 – Work on backlog	3 weeks	27-05-2024	14-06-2024
6 Sprint 5 – Work on backlog	3 weeks	17-06-2024	05-07-2024

4. Testing strategy and configuration management

4.1 Testing strategy

We will be working with the Test-Driven Development strategy, which means we will write unit tests before writing the classes and methods. For the backend we will use MSTest to write the unit tests. The front end will be tested using Cypress.



To ensure test quality we will be using Sonarqube. The testing will be automated in our CI/CD pipeline.

We will perform usability tests as often as possible by creating a demo at the end of each sprint. These demos will be presented to the client to get feedback. However, as we want the opinion of the end-user as well, the website will be QA (Quality Assurance) tested by the development team as well as users that have no existing knowledge of the project. This is to minimize bias testing due to knowing the expected behaviour. Additionally, there will be a test plan listing all features and if they are ready for testing.

4.2 Test environment and required resources.

We will set up our own CI/CD environment. When someone creates a merge request for a branch, the CI pipeline should be triggered to automatically run the tests. Once all the tests have passed, the code can be merged, and the CD pipeline should be triggered to deploy the changes.

4.3 Configuration management

We will be using the Fontys GIT for projects. We will be using the main branch, a development branch and feature branches for different type of features we are working on. Once the feature is implemented and evaluated, a merge request will be made. A team member that did not work on the feature reviews the code and either approves the merge request or tells the developer what to improve before. When the request is approved it will be merged into the development branch. No branches will be merged before the code is reviewed and all written unit tests pass.

5. Finances and risk

5.1 Project budget

The group has no budget to make expenses on themselves or as a group. They need to make use of free software and access to information such as licenses to the Quantum Chess game itself. Also, the whole team needs access to the GitLab environment that also can be accessed by the client. Furthermore, when the group do need access to certain paid software, they can ask Fontys for potential licenses.

5.2 Risk and mitigation

Risk	Prevention activities	Mitigation activities
1 Company contact gets sick or leaves the company		Contact the company for a new contact person
2 Laptop stops working	Have the data/code in the cloud	Get the last version from the cloud and continue there
3 Project member leaves the group	Know what your team is doing in the project and keep check with each other	Schedule a knowledge transfer and divide tasks
4 Security breach	Follow four eye principle to make sure code is clean and no issues are created	Inform the stakeholders and fix the issue
5 Time crunch	Ask for help from team members when you take too long with a task.	Inform the stakeholders and move to the next sprint
6 A member of the team is unable to meet the deadline for their work.	The team will split the work depending on the skills of each person in the group to ensure efficiency.	When possible, give the person extra time to finish their work. If this is not possible, someone else from the team will help resolve the problem.
7 When a team member is not respecting and following the main rules and agreements that the group has agreed and set up at the beginning of the project.	Every team member will need to approve the rules and agreements that are going to be set during the SCRUM meetings.	When a problem occurs, this will be approached by having a team talk. When this fails for any reason, the project owner and/or the semester coach will be informed and asked to act.