# *PLAN OF APPROACH*

## QUIZZION 2.0

*Project HBO-ICT Corp*

*Prepared for*

*Parantion*
*Keulenstraat 12, Deventer*

Document version: 0.1

*Prepared by*

*Team Dev-a6-1*
*Saxion University of Applied Sciences | 2020*

# Table of Contents

# 1 About the document

## 1.1 Version history

| Version | Updates | Author | Date |
|---|---|---|---|
| V 0.1.0 | Initial document structure | Tien | 04/05/2020 |
| V 0.1.1 | Filling in the empty sections | Tien, Vladyslav, Vy, Victor, Mark, Katarzyna | 09/05/2020 |
| V 0.1.2 | Adding 60% competence and updating requirement table | Tien, Vladyslav, Vy, Victor, Mark, Katarzyna | 12/05/2020 |
| V 0.1.3 | Updating user stories and backlog | Tien, Vladyslav, Vy, Victor, Mark, Katarzyna | 16/05/2020 |
| V 0.1.4 | Adding 20% competence | Vladyslav | 16/05/2020 |

## 1.2 Introduction

This document is the Project plan of approach for the Quizzion 2.0 web application built for client Parantion. The application is developed in cooperation with Saxion UAS Deventer in accordance to the project HBO-ICT Corp

This document describes the following:

1. Project overview, descriptions and requirements
2. Project plan and organization

# 2 About the project

## 2.1 Company background

This project will be developed on request of Parantion company.

This software company is placed in Deventer. Parantion has created 3 main projects:

- Easion is an online survey tool to create questionnaires.
- Growth document is a tool created for schools to track children`s study progress.
- Scorion is an online tool for higher educations to track student`s progress.

Parantion emphasizes on their safe storing of customer`s data which is only according to law.

## 2.2 Scope management

For this project, the responsibility of the deliverables will be held down on the **Product Owner**. The scope of this project is set up by the "**PARANTION**" company.

The team of developers will establish the documentation for the deliverables. All change requests will be submitted to the **Product Owner** who will then evaluate the requested scope change. Upon acceptance of the scope change request, the responsible person of the entire project (**Paul De Groot**) will accept the scope change request to the **Client** for deliverables.

Based on feedback and input from the **Client** and the **responsible person** of the project (Paul de Groot), further developing process will remain the same or will be changed according to the remarks.

### 2.2.1 Objectives

1. Create a quiz web application with several functionalities and features naming Quizzion 2.0 as an update of the current module Quizzion that the client's having
2. The client wants to see some ideas and usecases that they could apply Quizzion for (educational and recreational purposes)
3. Create a happy and fun quiz application for the end users to enjoy

### 2.2.2 Secondary objectives

1. The product is responsive through different screens size and devices
2. The product offers flexibility for users to customize their playgrounds
3. The client wants advices on using AWS for integration in this project

### 2.2.3 Out of scope

1. The admin feature to be implemented
2. Using data from other application like Easion or Quizzion 1.0

## 2.3 Requirement management

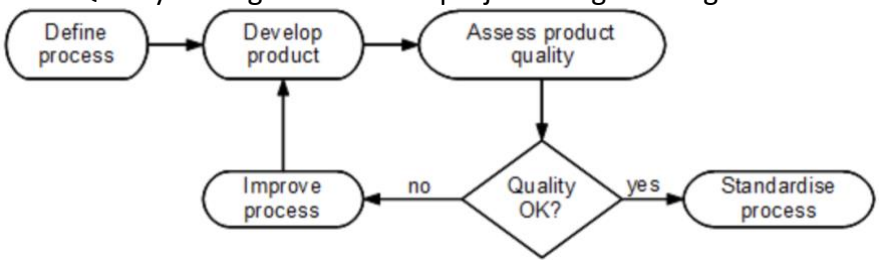
| Req.nr | Description | ISO 25010 | F/NF | Priority |
|---|---|---|---|---|
| 1 | The app includes standard functions: creating quizzes and inviting players | U/Functionality | F | MUST |
| 2 | The app is applicable in both formal and informal scenarios | B/Compatibility | NF | SHOULD |
| 3 | Create a safe architectural structure and define how it communicates with the client's standard platform/API | B/Portability | NF | MUST |
| 4 | The application is built with VueJS | B/Portability | NF | MUST |
| 5 | The application is built with Quasar framework | B/Portability | NF | MUST |
| 6 | A minimal version of product to be delivered and updated continuously | B/Portability | NF | COULD |
| 7 | Moderators are able to customize certain sections in the quiz | U/Functionality | F | SHOULD |
| 8 | Participants is able to view the quizzes' result | U/Functionality | F | MUST |
| 9 | Moderator is able to view the quizzes' result | U/Functionality | F | MUST |
| 10 | Moderator is able to customize the setting for viewing result on what participants can see | U/Functionality | F | COULD |
| 11 | The moderator has a dashboard with overview of quizzes and questions | U/Functionality | F | MUST |
| 12 | The moderator's dashboard shows the statistics of the players | U/Functionality | F | COULD |
| 13 | | U/Portability | F | MUST |

| | | | | |
|---|---|---|---|---|
| | Management and creation of the quizzes are done on desktop/laptop and tablet | | | |
| **14** | The naming conventions follow the client's wishes, referring to the "getting started" documentation | S/Maintainability | NF | MUST |
| **15** | The project structure realizes "Separation of concern" principle (referring to the "getting started" documentation) | S/Modularity | NF | SHOULD |
| **16** | A prop (referring to vuejs library) is declared with a required and default value | S/Maintainability | NF | COULD |
| **17** | The project uses Vuex for store and state management | S/Modularity | NF | MUST |
| **18** | Vuex store modules adhere the folder structure mentioned in the "getting started" doc | S/Modularity | NF | SHOULD |
| **19** | The participants join the quiz via link in text message | U/Functionality | F | MUST |
| **20** | A link is generated and copiable when the moderator clicks "invite players". | S/Functionality | NF | MUST |
| **21** | Application overall flow is: creation -> invitation -> answer -> view the result | U/Functionality | F | MUST |
| **22** | The flow of creating quizzes and questions feel natural | U/Usability | F | COULD |
| **23** | The moderator can create multiple choice questions | U/Functionality | F | MUST |
| **24** | The moderator is able to create different types of question | U/Functionality | F | COULD |
| **25** | | S/Functionality | NF | SHOULD |

| | | | | |
|---|---|---|---|---|
| | Using relational database if we need to store extra data besides using the client's current API | | | |
| **26** | The moderator has option to limit the number of participants | U/Functionality | F | WOULD |
| **27** | The participants can join the quizzes from a phone to a desktop | U/Portability | F | MUST |
| **28** | Deployment via docker container | S/Portability | NF | WOULD |
| **29** | Font families are Montserrat and Roboto | S/Usability | NF | SHOULD |
| **30** | The app uses the Parantion colors scheme | S/Usability | NF | SHOULD |
| **31** | The login functionality be available | U/ Functionality | F | MUST |

## 2.4 Quality management & deliverables

1. The Quality management of the project will go through the following process:




2. The quality of the product is reviewed every sprint. The test and review plan are as the sprint schedule (regarding article 3.1.3). Each developer in the team will be responsible to test and review a certain part of the product, the detail on this assignment will be updated later in the sprint process. After each developer finishes testing their part in building the product, our team will do a "cross review", meaning other team member will review each other's part. Once the process is done, the scrum master of the sprint will do an overall testing before hand over to the product owner.
3. Final deliverable: Will be updated once the client's answer in which form they want the product to be shipped.

## 2.5 Communication management

| Name | Role | Contact info |
|---|---|---|
| **Vladyslav Petrykov** | Product owner | 466157@student.saxion.nl |
| **Eleina Nieborg** | Contact person (client's side) | eleina.nieborg@parantion.nl<br>+31 (0) 570 23 45 67 |
| **Paul de Groot** | Project Coach | p.degroot@saxion.nl |
| **Kris Minkjan** | Project Coach (client's side) | kris.minkjan@parantion.nl |

## 2.6 Risks analysis

| Risk | How to manage the risk |
|---|---|
| The deployment of the system takes more time than it was assumed | Monitor the progress of existing projects |
| Identified new requirements in the last stage of the project. | Brainstorm all requirements during the Sprint 0, anticipate the worst-case or heaviest-use scenario |
| The client is not satisfied with the final version of the product | Stay in contact with the client and update the client with the progress |
| There is a problem with communication between the client and the product owner | If a meeting is impossible, reschedule for the next working day. |
| One of the team members will not be able to continue her/her work | Ensure that team members collaborate and share knowledge |
| Problem with requirements, not specific | Discuss all requirements together |
| Too many tasks to do at the end of the project | Set a realistic schedule |
| Spend too much time on extra/unnecessary features | Discuss every feature with other group members |
| Wrong priority of the task | Discuss all task with the team members |
| Problem with hardware | Be sure that the code in GitLab is updated, use backups |

# 3 Project schedule & organization

## 3.1 Realization of software competences

| Activities | Proficiency level description (level 2+) | Example of implementation |
|---|---|---|
| Implementation (18%) | Build and make available a software system that is comprised of several sub-systems while using existing components. | We are going to create 2 subsystems, one for client application and another for server to handle the application's needs. |
| | Integrate software components into an existing system whereby you safeguard the integrity, security and system performance. | We are going to Dockerize the deliverable, so the client will be able to run the project without any additional installation. |
| | Carry out, monitor and report on unit integration, regression and system tests, with attention for security aspects. | We are going to do the unit testing which will cover 100% lines of code, integration test which will verify that server and application work fine together and acceptance test by the client which will verify that we are creating the right software. |
| Design (14%) | Compile a design for a software system while taking into account the use of the existing components and libraries | Quasar framework based on Vuejs offer state of the art UI components and libraries which we can use design the frontend based on the mock-up. |
| | Apply design-quality criteria while taking into account security aspects and various types of devices. | Quasar framework supports desktop devices and mobile browsers (including iOS safari)<br>The mock-up is designed consistently so that the user interface will not be so different in different devices. |
| | Apply design-quality criteria while taking into account security aspects and various types of devices. | Quasar framework supports desktop devices and mobile browsers (including iOS safari)<br>The mock-up is designed consistently so that the user interface will not be so different in different devices. |
| | Create a design for a system that can process and consult a large quantity of data. | Data will be stored securely in a MySQL database. |
| | Record the quality of the design, for example by | Product will be tested in different devices. (desktop, mobile, tablet, etc.) |

| | | |
|---|---|---|
| | testing or prototyping, taking into account the formulated quality properties. | |
| | Compile test subjects according to a given test strategy. | Create some unit tests for fetching, updating and delete data in the database. |
| Management (10%) | Findings concerning the user perspective between the iterations in a design and development process. | In our case, “PARANTION” gave us a small set of rules that should be followed during front-end development. Such specifications will allow the development process to be faster. In all other cases, we should first be focused on the mock-ups and having them checked by Kris (our client) which of course will take time. |
| | Functional control processes. | Functional control process will be firstly reviewed by other team members and then will be checked by the AWS services by amazon. |
| | Management of new technological developments concerning the infrastructure. | Management process proceeds in the four steps:<br>• Planning<br>• Organizing<br>• Leading<br>• Controlling<br><br>The receipt to the client will be sent according to the hours that the entire team of developers spent to develop the application. The quality of the ICT services will be measured via AWS services. The contact person is always available during the working days so that client has always a clear vision of how the development is going. |
| | Methods and techniques to manage a software development process and safeguard the quality. | The process of the continuous integration is going to be done via ‘AWS’ service which will carry out the testing part of the web application so that, it will be available faster in the production. |
| | Given development environment, management and tests. | Given development: API from “PARANTION” with the manual how to use will be handled in the required part of the program according to the requirement management (2.4). Unfortunately, there’s no access to the source code of the server so that the quality can’t be properly measured. |

| | | |
|---|---|---|
| Analysis (9%) | Carry out an analysis for complex software-in-software systems including all non-functional requirements such as safety, security and privacy. | The requirements concerning privacy and security are mentioned in the requirement table:<br>Ex: The quiz moderator has to log in to see dashboard: which will be done using token, md5 hashing for security and privacy |
| | Carry out a requirement analysis for a software system with various stakeholders, while considering the quality properties including security. | The analysis of client's requirements that is reflected in the "requirement management" table<br>The analysis of different software use cases that is done in the use case diagram in the functional design |
| | Define acceptance criteria based on quality properties and a risk analysis carried out with, among others, attention for security aspects. | The definition of done with different criteria acceptance<br>Risk analysis can be found in the plan of approach |
| | Set up an acceptance test based on quality properties. | The test plan that would be included in the technical design later in sprint 1 |
| Advice (9%) | Provide advice on the purchase and selection of software components during the development of a software system whereby the cost aspect plays a role. | We could give advice on…<br>- The usage of AWS for some solutions.<br>- The setup of testing suite. |
| | Provide advice on a section of the architecture or a limited software system. | We could give advice on…<br>- Distribution.<br>Best practices in Quasar. |
| | Give advice on the use of prototypes in validating the requirements. | We could give advice on…<br>Steps they may want to consider after the project is done. |

## 3.2 Realization of other competences

### 3.2.1 User interaction

1. UI/UX – We analyzed the clients' wishes on achieving certain user's experiences, like focusing on users' happiness, making the participating process like a flow and easy to join, … The research on how to achieve these particular experiences was conducted by our team members. We then implemented this research into creating the user interface and a prototype to pitch to the client during the meeting (realization activity)

### 3.2.2 Infrastructure

1. Testing (AWS) - during the time when each of the team member will close an issue regarding the functionality of the application it will be tested via AWS services. (or entire application will be tested at the end of the development process).

2. Web Sockets – one of the issues provided by the client to see the data in real-time:
   update result of the players on the scoreboard of the quiz online, see new users.

## 3.3 Overview sprint structure

### 3.3.1 Sprint planning

In every sprint, we will have a sprint planning session in the beginning of the sprint. This meeting will be initiated by the product owner and continued by the scrum master of that sprint. According to the schedule, we will have 4 phases, starting on week 1, week 3, week 5 and week 7

1. Sprint 0 (Week 1+2): Preparation phase
2. Sprint 1 (Week 3+4): Realize Minimum Viable Product (MVP) – first release
3. Sprint 2 (Week 5+6): MVP refinement
4. Sprint (Week 7+8): Product polishing+ Final deliverable

### 3.3.2 Standups and team meetings

As part of the agile methodology, in a sprint, we have daily stand ups among the team members. This takes place in the beginning of every team meeting
Aim and purposes:
1. To maintain the close communication among team members
2. Identify problems or risks at an early stage
3. Figure out the solutions for current troubles that each member is having

### 3.3.3 Sprint review and deliverables deadline

| Turn-in points | Deadline |
|---|---|
| **Plan of approach** | 9:00 am - 18/05/2020 |
| **Snapshot phase 1** | 9:00 am – 02/06/2020 |
| **Snapshot phase 2** | 9:00 am – 15/06/2020 |
| **Final deliverable** | 9:00 am – 29/06/2020 |

## 3.4 Team member

| Member's name | Student number | Role |
|---|---|---|
| Vladyslav Petrykov | 461157 | Product owner |
| Katarzyna Strozykowska | 472064 | Developer |
| Mark Kravchuk | 460888 | Developer |
| Victor Hubbers | 469487 | Developer |
| Vy Tran | 463298 | Developer |
| Tien Thai | 460592 | Scrum master (sprint 0) |

## 3.5 Code of conduct

1. Attend every “HBO-ICT Corp” class hour scheduled by the project coach.
2. Communicate with team members via Microsoft Team and Whatsapp for any updates. Require to response to message as soon as possible and at most 1 day late.
3. Use of GitLab is mandatory.
4. Work at least 4 days = 36 hours per week, this includes the team meetings and class hours. Activities related to project must be registered through the timesheet in Microsoft Team.
5. All members are required to update their working hour in the sheet in the end of the week (before next Monday), after that the scrum master will check this entry.
6. If you have an emergency and cannot do your part of the work on time, contact the scrum master and product owner for solution. If you cannot attend a meeting, inform the team member at least 1 hour before the meeting
7. Help your group members if they are in trouble and ask for help if you need it.
8. Attendance is a key for contribution in working process, team member needs to inform the group when attendance is not possible. In case that the case of absence is not specified and/or the team was notified too late, the member in question is awarded 1 absence point. This will also be discussed with the team member.
9. Strikes may be given depending on a situation (e.g being constantly late, skipping the workshops), it needs to be decided by the whole group
10. In case of long period illness, the teachers should be notified as well. Group members together with the teachers will discuss further actions
11. Be respectful and patient to each other.
12. Meetings are held during the workshop as well as after classes (could varies from Monday to Thursday according to the needs)
13. Gitlab will be used for project management
14. If there are any complications or disagreements, the whole team must come together and address this matter.

## 3.6 Timesheet

Time sheets are stored in our Microsoft Team, under “Effort Registration” tab. Each member is able to access this file in order to add their working hours on the project. This time sheet is also accessible for the project coach to keep track of the process.

Only focused work hours are logged, meaning that the team member progressed in some way during the written time without distractions. This progress may be research, coding or documenting etc.

## 3.7 Definition of done

Our definition of done for each user story:

- The functionality is manually tested by at least one group member. This means that the acceptance criteria for each user story have to be met
- The user story is well documented
- The user story is confirmed by Product owner and Scrum master
- The code is clean and maintainable

## 3.8 User stories and Product backlogs

### 3.8.1 User stories

QM – Quiz Moderator
R – Respondent

| Id | As a | I want | So that | Acceptance criteria |
|---|---|---|---|---|
| US-1 | QM | To create a quiz on desktop or tablet | responders can play it. | Quiz is shown in dashboard/saved in database. |
| US-2 | QM | To edit an existing quiz | It is more customizable | The quiz is updated |
| US-3 | QM | To customize the quiz by choosing a theme's color | It is more customizable | The theme’s color changed. |
| US-4 | QM | To customize quiz by adding a company logo | It is more customizable | The logo changed. |
| US-5 | QM | To invite players to the quiz | They can play | Clicking a generated link |

| | | | | |
|---|---|---|---|---|
| | | | | allows you enter a quiz. |
| US-6 | QM | To add multiple choice questions to the quiz | It becomes more interesting for respondents. | New question is shown in quiz overview. |
| US-7 | QM | To set a timing for every question | It becomes more challenging. | Maximum time is stored in database. |
| US-8 | QM | To edit existing questions of the quiz | It is more customizable | Updated question is shown in quiz overview. |
| US-9 | QM | To see all my existing quizzes | So that I can check/review start the quiz and review each quiz | Showing the result of quiz |
| US-10 | QM | To see a result of the quiz | I know who won. | Top 3 is shown on podium screen. |
| US-11 | R | To see a result of the quiz | I know how I performed. | The result is shown |
| US-12 | QM | To have a dashboard with the statistics of the players | I know how people performed | The result is shown |
| US-13 | QM | To see the answers of the players | Common mistakes for questions | The overview of all answers is shown |
| US-14 | R | To play quiz on different devices from mobile to desktop | The quiz is accessible for more people | App runs on both platforms and is user friendly |
| US-15 | QM | Customize the results that respondents see | The quiz is applicable in different scenarios | Quiz overview includes different settings |
| US-16 | QM | To add different types of questions | The quiz becomes more interesting & is applicable for different scenarios | New question is shown in quiz overview. |

| | | | | |
|---|---|---|---|---|
| US-17 | QM | To login with Quizzion account | Manage the quizzes | Login page redirect to dashboard when valid credentials are entered |
| US-18 | QM | To be able to limit the amount of respondents | I can control the sample size | Option for limiting number of respondents is in quiz overview |

### 3.8.2 Product backlogs

All the user stories were prioritized according to the 'MoSCoW' criteria.
*The estimation time is subjected to change (+-10 hours.) – Total: 530 hours*

| Story # | Story | Priority | Estimates (hours) |
|---|---|---|---|
| US-1 | Create a quiz on both laptops and mobiles | MUST | 35 |
| US-9 | Create a quiz dashboard | MUST | 30 |
| US-5 | Implement invitation of players to the quiz | MUST | 25 |
| US-14 | Implement answering functionality | MUST | 30 |
| US-13 | Create a quiz results table for quiz masters | MUST | 25 |
| US-17 | Implement login functionality | MUST | 25 |
| US-2 | Implement editing functionality for quiz name & description | MUST | 30 |
| US-6 | Implement adding of multiple-choice questions | MUST | 30 |
| US-8 | Implement editing of existing questions | MUST | 25 |
| US-11 | | MUST | 30 |

| | | | |
|---|---|---|---|
| | Create a quiz results table for respondents | | |
| US-3 | Implement editing functionality for quiz theme color | SHOULD | 30 |
| US-4 | Implement editing functionality for quiz logo | SHOULD | 35 |
| US-16 | Implement adding of various types of questions | COULD | 30 |
| US-15 | Implement option to customize the results that respondents see | COULD | 40 |
| US-12 | Create statistics dashboard | COULD | 35 |
| US-10 | Create podium screen | COULD | 20 |
| US-7 | Implement timing functionality | WOULD | 30 |
| US-18 | Implement maximum number of respondents | WOULD | 25 |