Software Project Management Plan

Lezioni alla pari

April 19, 2020

Team Members

Ovidiu Andrioaia
David Cirdan
Luciano Mateias
Zhiyang Xia

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	04/20/2020	Initial release

Document storage

This document is stored in the project's GIT repository at: https://github.com/KilliKrate/Software-Documentation-G6/blob/master/docs/Software%20Project%20Management%20Plan/index.md

Document Owner

Group 6 is responsible for developing and maintaining this document.

Table of Contents

Overview

Purpose and Scope

Lezioni alla Pari is an online desktop application designed to streamline knowledge sharing between people inside a closed environment: any group can use our platform to create lessons in the form of illustrated documents or video, in order to instruct their peers on subjects of interest. Quizzes are also available as a tool to summarize (in the case of a company) or test (in the case of a class) the understanding of the subject.

Anyone can use this application: from sports groups and classrooms, to project teams and companies, the application uses a simple and straightforward structure so that anyone can pick up the program and immediately start learning without the need to check any user guide or manual. This is particularly important because this application is meant to cater to a wide demographic: a kid in a classroom should be able to use the program as well as a new intern in a big company.

The software will allow people to create a course, which will contain lessons on a set of topics. The authors of a course can decide to invite people as collaborators, which will actively populate the course and work together with the author, or a student, an individual that can only read and watch lessons. Automated quizzes can also be created and completed, and the author and collaborators will be able to check the answers given if needed.

Lezioni alla Pari will run on a client-server structure, which thanks to a distributed server system will allow us to deliver content quickly, and is easily scalable if necessary.

A mobile version of the program is feasible and will be delivered if the project is successful, and enough people start using it to justify cost and developement times.

Goals and Objectives

The main objective of Lezioni alla Pari is to give the members of a group a way to easily share knowledge and information between each other.

Project Goals:

1. Deliver an easy to use application that democratizes knowledge sharing in a way never seen before.
2. Give ITI G. Marconi, our main partner, prestige and visibility in the local region, if not nationally.
3. Deepen our knowledge of software engineering and architecture design.

Project Objectives:

1. Create a desktop client that allows people to create content, and delivers that content on-demand securely and promptly.
2. Create an easy to use and intuitive User Interface, and refine the User Experience in a way that all demographics can take advantage of the app.
3. Provide a way for authors to summarize information through automated quizzes, and give the ability to the former to review the answers of each participant.

Project Deliverables

Date	Deliverable
05/13/2019	Software Documentation
05/30/2019	End of first iteration: video streaming, database structure and partial back-end developement
06/16/2019	End of second iteration: back-end completion, front-end developement
06/17/2019	Beta release for stakeholders
09/13/2019	Application fully downloadable from Github repository

Assumptions and Constraints

Assumptions

1. Github will allow the team to collaborate on the project remotely by scheduling tasks and committing code

Constraints

1. The project will have to be supported by at least 3 partners before the beta phase, in order to have an acceptable amount feedback.
2. The application must satisfy security and privacy standards of public institutions before release, in order to guarantee that sensible information will not be made public through exploitation of weaknesses or hacking.

Schedule and Budget Summary

Cost Estimate

- 1 project manager and back-end developer and at 4 hours per week for 24 weeks
  96 hours * €20/hr = €1920

- 1 marketing specialist and front-end developer at 4 hours per week for 24 weeks
  96 hours * €20/hr = €1920

- 1 back-end dev at 4 hours per week each for 24 weeks
  96 hours * €15/hr = €1440
- 1 front-end dev at 4 hours per week each for 24 weeks
  96 hours * €15/hr = €1440

384 hours total, €6720 total, avg, €17,5 per hour

Schedule Summary

Date	Review / Milestone
04/10/2019	Project start
04/15/2019	Software Documentation and planning completed
04/16/2019	Start of the first iteration
05/20/2019	Review of the first iteration
05/22/2019	Start of the second iteration
07/01/2019	Beta release and feedback from stakeholders
07/08/2019	Further development of the platform
09/13/2019	Application fully downloadable from Github repository

Success Criteria

At least 3 official insitutions/companies that have successfully adopted our platform after release will be considered as a success to the project

Definitions

- **Use Case**: Describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.
- **Back-end**: The part of the software that handles the logical part of the application (fetching and saving data from database, sending changes, etc.)
- **Front-end**: The main user interface of the software, through which the user interacts
- **Appplication**: Lezioni alla Pari, which is the software that is being developed, specified in this document.
- **Project**: Activities that will lead to the production of the application described in this document.
- **User**: A person who uses our application.
- **Client-Server**: a model used for the developement of internet-based applications.
- **Client**: The part of the software that is used by the user.

Startup Plan

Team Organization

Team members:

- Project Manager :
    - Ovidiu Costin Andrioaia.
      The project leader is responsible for creating the project plan, managing risks, and running the weekly team meeting.
- Front-end developer:
    - David Constantin Cirdan - Zhiyang Angelo Xia.
      The Front-end developers are responsible for designing and and implementing the application's UI.
- Back-end developer:
    - Ovidiu Costin Andrioaia - Luciano Mateias.
      The Back-end developer is responsible for coding the logic of the application.
- Marketing :
    - Zhiyang Angelo Xia.
      Deals with the sale policies of the company's products and establishes the techniques and strategies that must be adopted in order to increase sales.

Project Communications

The information is collected in the weekly meeting, in which each member reports on the obtained feedback, so as to be able to obtain a concise summary and check the progress of the project. The feedback reporting occurs thanks to the use of multiple channels:

- Social media
- Github
- Referral through users

Technical Process

As a development methodology the team chose to use SCRUM together with the PEP-8 code standard. SCRUM enables the team to develop the software quickly and with the assistance of feedback for further iteration.

Tools

Tools used for development:

- Programming Language: Python - JavaScript
- Version Control: Github
- Defect tracking: Bugzilla

- Build tools: PyCharm
- Database: SQL

Work Plan

Resource Estimate

The following estimates of resource spending are based on time, estimated effort, actual effort and the dependecies between the tasks:

> *Software documentation:*
>
> > *Team members involved: all the team takes part Time: this task should be done in three days Estimated effort: substantial Actual effort: substantial Dependencies: all the later tasks will be influenced by this one*
>
> *Find partners:*
>
> > *Team members involved: all the team will take part to a certain extent, but the marketing specialist will be in charge of decision making. Time: the task is estimated to take up to one month before conclusion. Estimated effort: major Actual effort: major Dependencies: all the tasks regarding the advertising and launch of the platform will be influenced*
>
> *Database architecture design:*
>
> > *Team members involved: all the team Time: up to one week Estimated effort: average Actual effort: average Dependencies: software development tasks*
>
> *Video streaming architecure design:*
>
> > *Team members involved: all the team Time: up to one week Estimated effort: substantial Actual effort: substantial Dependencies: software development tasks*
>
> *Back-end architecure design:*
>
> > *Team members involved: back-end developer and project leader Time: up to one week Estimated effort: substantial Actual effort: substantial Dependencies:*
>
> *Database development:*
>
> > *Team members involved: all the team Time: up to two weeks Estimated effort: average Actual effort: average Dependencies: all the software development tasks*
>
> *Back-end development:*
>
> > *Team members involved: back-end developer and project leader Time: up to two months Estimated effort: average Actual effort: average Dependencies: front-end tasks and overall software constraints*
>
> *Client design and development:*
>
> > *Team members involved: front-end developers Time: up to one month Estimated effort: average Actual effort: average Dependencies: appeal to the partners and stakeholders*
>
> *Private Beta release:*

> > *Team members involved: all the team*
> > *Time: one month from the end of the project Estimated effort: major Actual effort: major Dependencies: market contribution and reviews of the product*
>
> *Partner and stakeholder feedback:*
>
> > *Team members involved: all the team Time: up to one week after beta release Estimated effort: minor Actual effort: minor Dependencies: changes on the software on a structural and esthetical level*
>
> *Debugging:*
>
> > *Team members involved: all the team Time: up to two weeks Estimated effort: average Actual effort: average Dependencies: a well functioning software for the future*
>
> *UI and UX improvements:*
>
> > *Team members involved: all the team Time: up to two weeks Estimated effort: average*
> > *Actual effort: average Dependencies: a well functioning software for the future*
>
> *Final release:*
>
> > *Team members involved: all the team Time: roughly six months after project start Estimated effort: average Actual effort: average Dependencies: none*

Release Plan

Iteration #1

- Summary: design all the application architecure and find partners

Features / Deliverables	Estimated Effort	Actual Effort
Architecture design	Substantial	Substantial
Finding partners	Major	Major

Iteration #2

- Summary: development of the database, front-end and back-end with beta release

Features / Deliverables	Estimated Effort	Actual Effort
DB, Back / Front-end	Substantial	Substantial
Beta release	Major	Major

Iteration #3

- Summary: final release, with the approval of partners, stakeholders and various software debugging

Features / Deliverables	Estimated Effort	Actual Effort
Final release	Average	Substantial

Iteration Plans

- **First iteration**: intention agreement will all the team on how the software should be developed, starting to search for partners interested in the project while developing the overall design and framework. In the meantime devs will also develop and deploy the video streaming architecture, the database structure and will deploy a part of the back-end server logic.
- **Second iteration**: front-end design and implementation and back-end completion. After the core application is finished, the private beta will be launched for all partners to use.
- **Third iteration**: past the beta and the general feedback on the software, start fixing bugs and improve the platform, at last launch the final release.

Control Plan

Monitoring and Control

The following list of dates includes formal reviews of the project and major accomplished milestones in reference to the established schedule.

Date	Deliverable
04/15/2019	Software Documentation
05/20/2019	End of first iteration
06/30/2019	End of second iteration
07/01/2019	Beta release for stakeholders
09/13/2019	Application fully downloadable from Github repository

Configuration Management Plan

This is the procedure to follow when making changes to the final product:

1. All the data is stored on a GIT repository hosted by Github, this is the main container.
2. All the changes to documents and files are tracked via commits and comments inside them, including the software documentation.
3. Changes are verified only after all the development team has seen and approved them.
4. The change control procedure is the one that follows:
    - send an e-mail to one of the developers, where it is explicitly described the parts of the software to modify, how impactful this changes are, the needed resources and time constraints.
    - if the e-mail doesn't receive any response within 30 days, the request will be dispatched, otherwise the recipient will contact the applicant and set the application process.
    - the petitioner from now on will be part of the project development and will implement the new feature with close collaboration with the team,

following all the assigned instructions.

- after the change has been successfully actualized, all the documentation and schedule will be updated with the new changes, as well as the public software.

Supporting Process Plans

Risk Management Plan

Rank	Risk	Probability of loss	Size of loss	Risk exposure	Response
1	Schedule	Likely	Major	High	Adhere to the schedule
2	Dev Team’s lack of relevant experience	Likely	Major	High	Let the dev team spend some time learning the needed technologies
3	End of financial support	Likely	Moderate – Major	Moderate	Constantly check for new fund opportunities
4	End of support for project requirements	Likely	Moderate	Minor	Talk to the managers of each of the library/framework used to ensure they will get updated till the project is released
5	Commerce team’s inexperience with ads	Unlikely	Minor - Moderate	Minor - Moderate	Let the commerce team spend some time learning the usage of Google AdSense
6	Customer change of idea	Unlikely	Minor	Moderate	Illustrate each step to complete the project and ensure the customer agrees to every revision made during the project (Agile method)

Test Plan

The test plan defines the items that will be tested, methods for testing, and a schedule detailing the tasks, owners, and time line.

The test plan will be available in a separate document in the version control system at: https://github.com/KilliKrate/Software-Documentation-G6/blob/master/docs/Test%20Plan/index.md

Product Acceptance Plan

At the completion of each iteration, the prototype will be tested. Testing will be made via Travis CI and Docker to create an isolated and uncontaminated environment. The software tested on Docker will face multiple checks with different operative systems to ensure the proper behavior.