Project Plan
AtlasBot

Version 1.0

Prepared by:

Jonathas Felipe da Silva

03 Sep 2019

Revision History

Author	Date	Reason For Changes	Version
Jonathas F Silva	27/08/19	Bot prototype	1.0
Jonathas F Silva	03/09/19	Project Plan	1.1

Table of Content

1. Project Overview

Today we use some kind of artificial intelligence daily, these systems are the basis of many services that we use and often do not even realize. AI artificial intelligence has ceased to be considered a work of fiction for some time and is increasingly present in our daily lives. Integrating an AI into predictive systems, smart cars, face recognition, data security, financial markets, healthcare, referral systems, Google, and Natural Language Processing (NLP) is very common today. In this last example, natural language algorithms help call center systems identify fast routes to provide the information the customer needs. It's the end of endless menus until you reach the consumer's goal. Further narrowing the project's point of interest is bots or chatbots, where AI is used to focus on NLP to talk to a person, and to help them or to propose some information that is useful or essential to the user.

1.1. Project scope

The history of natural language processing (NLP) generally started in the 1950s, although work can be found from earlier periods. In 1950, Alan Turing published an article titled "Computing Machinery and Intelligence" which proposed what is now called the Turing test as a criterion of intelligence. The project focuses on NLP and specifically bot or chatbot, which we call AtlasBot, where it is able to interact with the user and answer their questions and pass on information the user needs, AtlasBot will still be able to learn from the user, even questions he has never heard before. AtlasBot can control your home with day-to-day functions such as turning your home lights on or off, checking if the light is on, information about your home's entrance or door. All this by talking to the bot.

1.2. Major software functions

The online peer evaluation system will be divided into four parts:

- Conversation
- Pass information
- Learn
- Smart home

1.2.1 Conversation

The conversation with the bot happens through a chatbot in which the messages are exchanged in writing.

1.2.2 Pass information

AtlasBot is capable of transmitting information through conversations, either if the user asks or if he suggests information that the user would probably need.

1.2.3 Learn

learn your habits and already automate some functions.

1.2.4 Smart home

AtlasBot will be able to turn your home into a smart home by automating some basic everyday functions

1.3. Deliverables

Sprint1	Prototype Bot
Sprint2	Project Plan

2. Management Structure

2.1. Project Lifecycle

The life of the project revolves around the development and implementation and the creation of the structure that is physical, ie in hardware..

2.2. Project Organization

2.2.1 External Interfaces

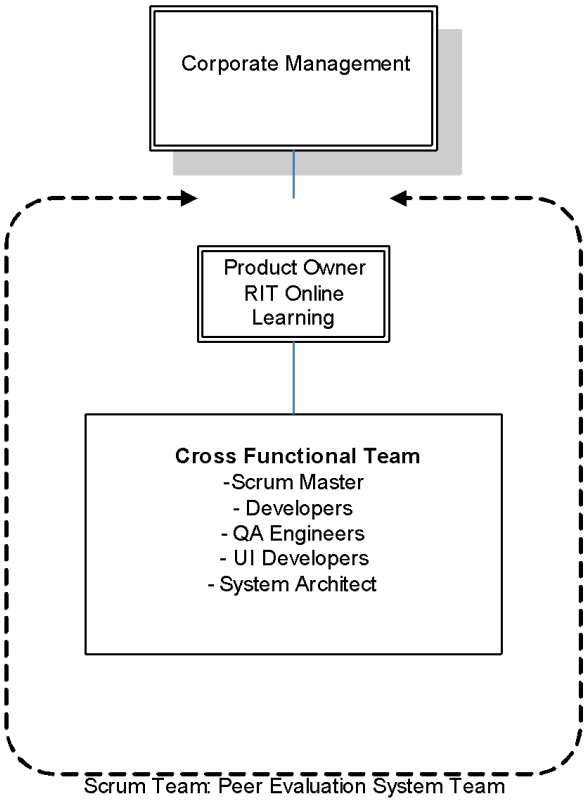

Figure 1.0, External Structure

2.2.2 Internal Structure

This project can be completed with a team of four people in a period of 15 weeks. The team will consist of software engineers capable of performing various tasks starting from requirements gathering, designing, database management, developing, testing to management. All the members will rotate between different roles to the final product delivered.

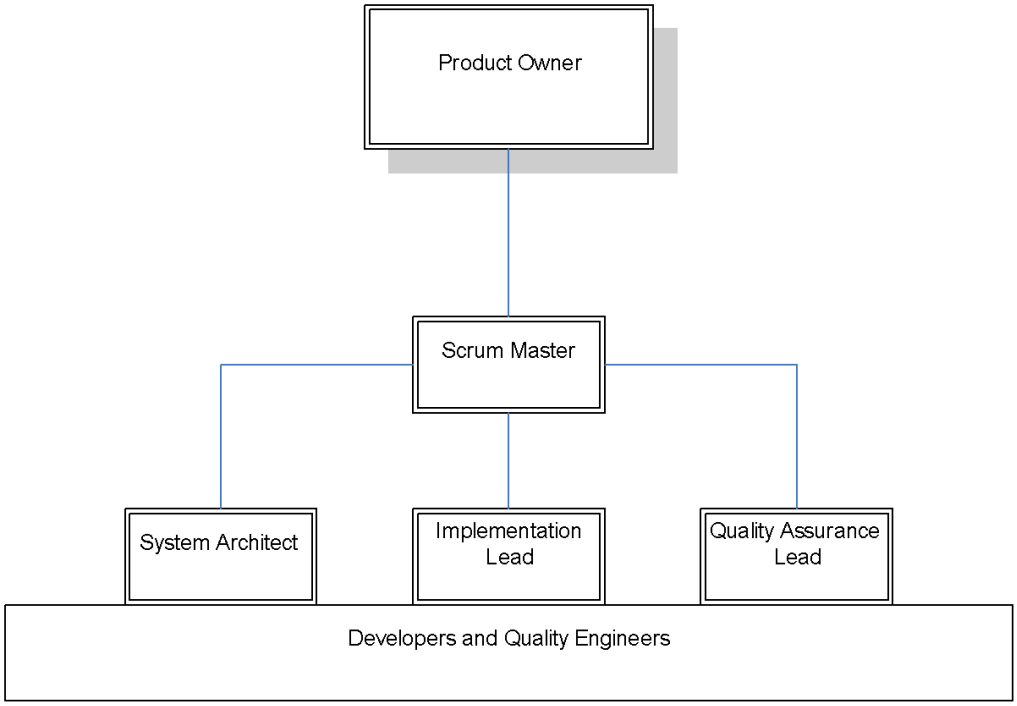

Figure 1.1, Internal Structure

2.2.3 Roles and Responsibility

Role	Responsibility
Scrum Master	Primary purpose is to mitigate project issues that may impede progress as they arise. Also, due to limited resources the Scrum Master will also be responsible for implementing the Peer evaluation system.
Product Owner	Control the budget and resources allocated to the project.
Project Team	Implement the goals and ideas of the project

2.2.4 Staffing

The project will be completed with a team of four members in a period of 15 weeks.

The team will include the following members:
1 Scrum Master (Developer)
3 Developers

2.3.

Risk Management

2.3.1 Top Risks List

1. New Technologies
2. myCourses Integration
3. Project quality will be unacceptable due to vague ambiguous requirements
4. Project will take more time for the team to complete than the amount of time budgeted
5. Changing requirements will lead to extreme feature creep with each iteration
6. Project is not feasible with given conditions of schedule and resources.

Risk	Probability	Impact	Exposure
1	0.20	4	0.80
2	0.25	4	1.00
3	.25	4	1.00
4	.20	4	0.80
5	.20	4	0.80
6	.05	5	0.25

2.3.2 Risk Mitigation Strategy:

1. The more experienced team members will try and educate/ guild other team members with new technologies
2. The team members new to technologies will start with building small prototypes on their own to grasp new technology
3. Specific tasks will be assigned to specific members based of sill set
4. Help of domain experts will be taken to address integration with myCourses
5. Through many sprints the requirements will be revised many times to ensure that clarity is achieved.
6. By using past projects/experience as a way of estimating effective effort, a better approximation of the team's abilities can be made. This would allow for a more accurate range of justifiable schedule deadlines.
7. Throughout every sprint, each decision will be evaluated to make sure that it aligns with the overall goals of the project.
8. The team will undergo many team building events as deemed necessary by the team.

3.

Planning and Control

3.1. Estimation Method

Schedule oriented practices will be used to deliver the final product in the span of 20 weeks. The customer will set the release date. The customer and the development team will then come up with an agree set of functionalities to be delivered on the release date. The customer prioritizes the list of features according to their needs and the development team will provide the estimates for implementing the features.

3.2. Resource Identification

3.2.1 Staff

The team size will be constant throughout the project.

3.2.2 Time

Start Date: Nov 28th, 2005
End Date: May 19, 2006

The final released date is set to second week of April, 2006. Features are to be delivered iteratively every one to four weeks, depending on the complexity of the implementing feature.

3.2.3 Cost

N/A

3.3.

Resource allocation

3.3.1 Work Breakdown Structure

Detailed Work Breakdown Structure will be created after requirements gathering. Figure 1.2 shows the product based Work Breakdown Structure.

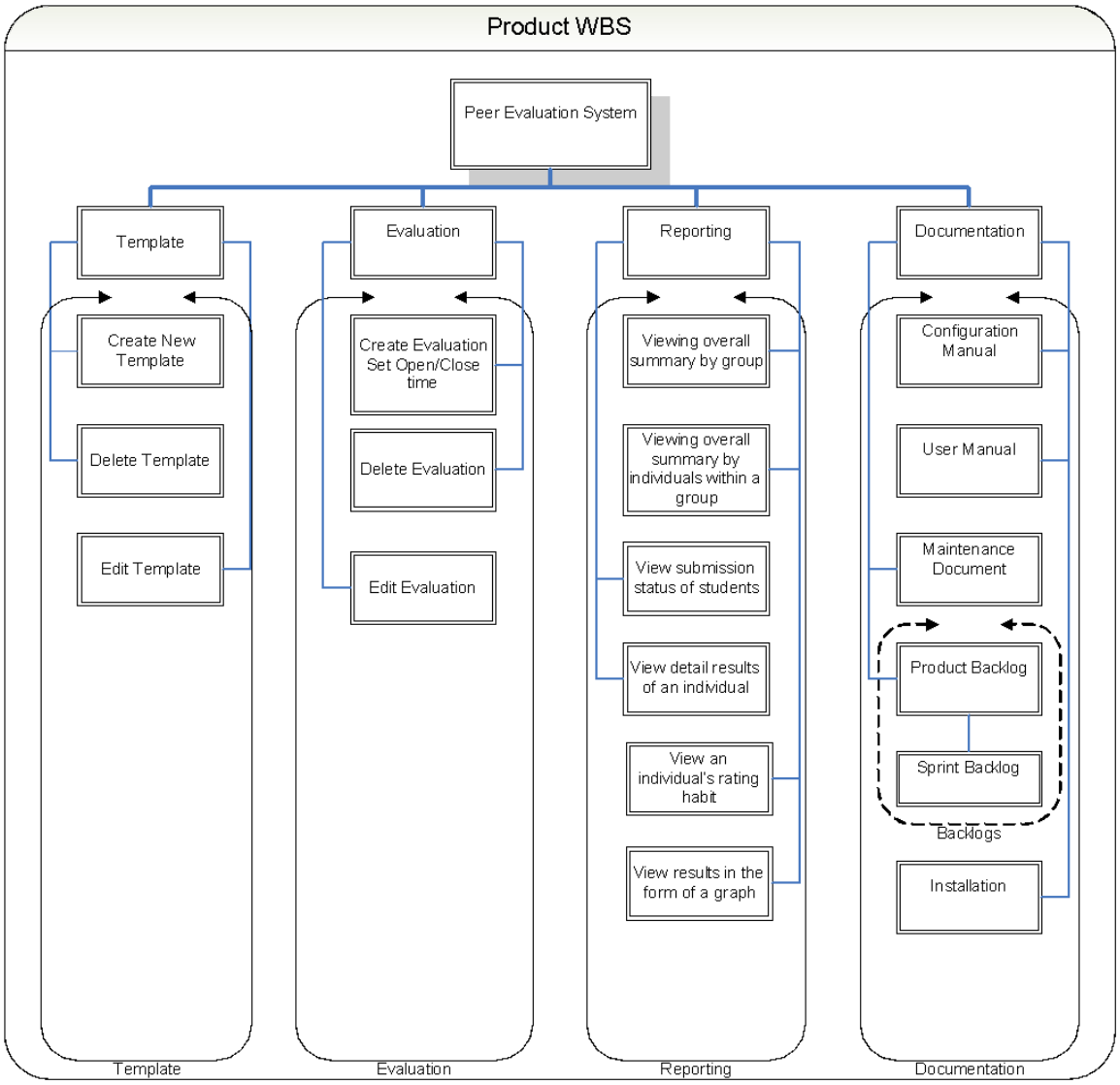

Figure 1.2, Work Breakdown Structure (Product Based)

3.3.2 Schedule

For detailed schedule refer to the Senior Project Plan.mpp.

3.4. Tracking and control

The Scrum Master/Development Team oversees all the Scrum meetings and tracks the development status. The development team identifies the initial backlog at the start of

the sprint, and periodically tracks the progress made by the development team. Progress tracking is done during the Scrum meetings, where team members are required to state their development progress relative to the sprint backlog.

To improve the process of identifying the number tasks to complete for a particular sprint and the length of the sprints, metrics that address the questions of "how many tasks should be associated for each sprint?" and "how long should a sprint last for" will be utilize. The selected metrics should assist the Scrum Master/Development Team in writing up sprint backlogs.

The following metrics will be implemented in the development of the product:

- Number of tasks completed for a particular sprit (Work effort distributed for each sprint)
- Number of open bugs
- Total numbers of hours
- Total number of bugs
    - Per feature
    - Per task
    - Bug type
        - UI
        - Feature
        - Feedback
- Sprint backlogs
- Product backlogs

3.4.1 Milestones

Major Milestones

The final product release is on third week of May, 2006.

Minor Milestones

These milestones are the iterative and incremental feature delivery at the end of each sprint. This allows customers to see the progress made compare to the overall project. This also allows customers to give feedbacks on the developed features, which can then lead to changes or additional features to be implemented in the next sprint.

3.4.2

Reporting

Weekly Report

As mentioned above, team members are required to give a quick status of their weekly progress relative the expected sprint delivery.

Sprint Report

The Scrum Master/Development Team is responsible for creating and delivering sprint information (sprint backlog) to the project team and the stakeholders. At the end of the sprint, the project team meets with the stakeholders where they will see the developed functionalities. Base on this delivery, features to be addressed in next sprint is re-defined. From the meeting and the backlog, the stakeholders can visualize the development progress of project.

3.4.3 Estimation Refinements

After each sprint, any of the pre-defined features can be changed. Depending on customers' inputs, features can be added, removed, or re-prioritized. Estimates are also refined or redefined based on these changes. Obstacles and problems encountered in the previous sprints are taken into account when refining the new estimates.

4. Technical Process

4.1. Methods, Tools and Techniques

Scrum Methodology: The whole project will be delivered in chunks depending on the requirements prioritization. Development team will get a specific amount of time to implement each chunk of functionality.

Revision Control: CVS will be used to track changes to all documents, source code, and any other relevant files.

Daily Builds: Source code will be build every night to avoid integration issues.

4.2.

Technology

4.2.1 Environment

The web-client shall be accessible via http request forwarded through any of the supported browsers listed below:

- Mozilla Firefox 1.07 or higher
- Microsoft Internet Explorer 6.0 or higher
- Netscape Navigator 7.0 or higher

4.2.2 Methods, Tools and Techniques

Software Tools:

- .NET 2003 Framework
- Microsoft Project will be used to track overall progress.
- Microsoft Office 2003
- MS-SQL Server 2000
- IIS Web Server 6.0
- Trac (Integrated SCM and Project Management)
- Subversion Tracking System

4.3.

Project Artifacts

Document Name	Description
Project Outline	A general overview of the project with some of its key features.
Project Plan	A document containing the following subsections: ● Project Scope ● Management Structure ● Planning and Control ● Technical Process ● Supporting Plans
Requirements	A formal requirements document containing User Stories, Use case realizations etc.
Project Schedule (Sprint Schedule)	The breakdown of how long each iteration will take.
Source Code	Source code required to build an executable product and incorporate the solution with deployment tools on the client- and server-side platforms.
User Manual/ Setup Documents	Formal setup documents and User manual will be provided to help users in setup and maintenance of the product. (User Manual will be optional and will only be provided if time permits)
Design / Feature Documents	Design and Feature documents will be provided for maintenance purposes.