Software Project Management Plan

ETL Support 2019 - Group 2

January 7, 2019

Team Members

- Elia Vicentini (Scrum Master)
- Simone Gandini
- Luca Landolfo
- Enrico Marocchio
- Adrian Munteanu

Document Storage

This document is stored in a GitHub repository, where you can also find all the other types of documentation regarding this project.

Document Owner

Luca Landolfo is responsible for the development and maintenance of this document.

Table of Contents

---

1. Overview

1.1 Purpose and Scope

The team behind ETL has the task of providing a script to **manage, modify, update and provide a backup plan,** for the data obtained from another project in our working environment, MarconiTT, a web application.

This script has the task of speeding up, supporting and updating the data needed throughout the web app process.

The script, written entirely in *Python 3.7*, communicates and works with different *SQL* databases within the work environment by **moving, deleting and updating** data from one database to another.

You will need to start the Script via any shell, from *cmd* (Windows) to any *Linux* or *OSX* command prompt.
You will be able to start the program in *two* different ways:

- **direct**
  - The program will connect **directly** to the school's main database in order to work faster.
- **api**
  - The program will connect to the school's database **via an API**, specifically created by the MarconiTT team.

The program will be used primarily by the authorized stakeholder: ***Prof. Bellini G.*** (RefOrarioWeb).

1.2 Goals and Objectives

The goal is to provide a functional and fast script, for a better management and reliability of the **data concerning the school's classroom reservations**, which will then be used by the MarconiTT web application for a better visualization of the data.

Project Goals and Objectives:

1. Create an app that is as fast as possible in calculations and simple to use.
2. Speed up the process behind the school's classroom reservations.
3. Provide correct and reliable data to the MarconiTT team.
4. Learn more about SQL, Databases and Python.

1.3 Project Deliverables

Date	Deliverable
29/03/2019	Plan *exportdata.py*
29/03/2019	json configuration file creation
30/03/2019	Production *setup.py*

02/04/2019	Prototype of *exportdata.py*
02/04/2019	Tests
02/04/2019	Bug fixes
03/04/2019	Script design for log files
04/04/2019	Completed the first version of the program
20/09/2019	Update *exportdata.py*
30/09/2019	Bug fixes
30/09/2019	Update all scripts
21/10/2019	Finished the script for log files
16/12/2019	Improvements on *exportdata.py*
16/12/2019	Finished the final version of the product

1.4 Assumptions and Constraints

- ***Assumptions***
    - The data is already correct, available and easily accessible.
    - The machine on which the program will be started already owns Python 3.7.
    - That all databases are available and connected.
- ***Constraints***
    - Stakeholder approval required for the execution of SQL queries.
    - Can only work at school.
    - Not much time.

1.5 Success Criteria

Creation of a working, secure and fast running script, runnable everywhere and easy to use, that not only updates the data on the different databases, but also backs them up in case something goes wrong.

1.6 Definitions

Term	Definition
Actor	User or other software system that receives value from a user case

Developer	The person or team who is developing the application
Scenario	One path through a user case
Stakeholder	Anyone of the entities directly or indirectly involved in the project and its outcomes
User	The person or group of people that will actually use with the application
User case	Describes how an actor will perform tasks on the system. It may define many different variants called scenarios that can lead to different outcomes.

2 Startup Plan

2.1 Team Organization

Role	Actor(s)	Responsibility
Scrum Master	Vicentini	Responsible for managing and coordinating the exchange of information between team members, also assign tasks to teammates.
Developer	Landolfo, Vicentini, Munteanu, Marocchio, Gandini	Write the software based on requirement and architect specifications
Tester	Gandini, Landolfo	Test the code, report issues, find and fix bugs, work on the code’s efficiency
Architect	Vicentini	Specify overall internal workings of application
Requirement Engineer	Munteanu, Landolfo, Vicentini	Define, document and keep track of the requirements

2.2 Project Communications

Event	Information	Audience	Format	Frequency

Team Meeting	Task status: what has been completed since last meeting and what were the problems encountered; planning on what's next.	All team members	Informal meetings before working; Messages for status updates and problems as they occur.	As needed
Project Status Report	Review of the sprint, status of the prototype, coding issues, problems relating the project	All team members, Stakeholder	Informal meetings before working	Iteration Closeout

2.3 Technical Process

Planned an incremental development process. The first iterations will be to lay the groundwork on what we need. Subsequent iterations will mostly work on the actual data.
Feedback from the Stakeholder will be used to improve the next iteration.

2.4 Tools

- *Programming Languages* - Python, SQL
- *Operating System* - Windows
- *Version Control System* - Git, the project will be stored on a GitHub repository
- *Development Tools*- Visual Studio Code, SQL Server

3 Work Plan

3.1 Resource Estimate

The following table shows a more or less accurate representation of how much time (in hours) *we expected to finish the project* (Effort Estimates), and how much time *we realistically took to complete the program* (Actual Effort).

Task	**Owner**	**Effort Estimates (Hr)**	**Actual Effort (Hr)**
Plan the project	Whole team	5	2.45
Config of the json file	Landolfo	0.20	0.30

Production *setup.py*	Munteanu, Vicentini	5	6
Creation of the 1st version of *exportdata.py*, with tests and bug fixes	Whole team	20	14
Script that works on log files	Marocchio	10	20
Update the program to work with MarconiTT's API	Whole team	3	1.30
Completed and Updated the script for the log files	Marocchio	2	2.15
Final improvements and bug fixes of the main program	Whole team	10	4
Total:	//	**55.2**	**50.2**

3.2 Release Plan

```
Iteration #1
29/03/19 - 04/04/19
```

Summary: First working prototype of the program, that connects to the server via a json config file and works with a basic script for writing logs.

Feature/Deliverables	Estimated Effort	Actual Effort
Plan the project	10	6
Complete 1st working prototype	30	30

```
 Iteration #2
20/09/19 - 16/12/19
```

Summary: Update the code to work with MarconiTT's API, in order to connect to the server. Fixed minor bugs and errors from previous version. Completed the script that works with log files.

Feature/Deliverables	Estimated Effort	Actual Effort
Update the main program so it connects to MarconiTT's API	3	1.30

Complete log files script	7	12.15
Recognize and fix the faults	10	4

3.3 Iteration Plans

3.3.1 First Iteration

The first iteration has as its intent to create a 1° working prototype of the program. It must be able to connect and perform the first simple operations on the databases, no matter how efficient. Also in this first iteration, we try to process a beta version for a script that will create and store log files for the main tasks that the program executes.

3.3.2 Second Iteration

This version of code will first have to work with MarconiTT's API to connect to the server and the databases. Obviously the connection method of the last iteration will have to work without problems anyway. The program will also need to be much more efficient and for this reason it will mainly be necessary to change the queries that are executed on the database.

4 Control Plan

4.1 Configuration Management Plan

The following procedure is to be used when making changes to all baselined work products:

1. All project work products will be stored in a GitHub repository.
2. All the documentation will be stored in a separated repository.
3. The change control procedure once a product is baselined is:
   - anyone wanting to make a change to a baselined item should fork the repo and once the changes have been made, a pull request can be opened, and if accepted, the changes will be pushed to the main codebase.
   - If the pull request is denied, the team will inform you on how the changes can be improved to work with the whole project.

5 Product Acceptance Plan

At the conclusion of each iteration, the version of the prototype has to be tested in an environment similar to the one used in our working space. This is necessary since the program has to work with multiple servers and databases, which all have different types of data.