Eureka Labs

Software Development Plan

Version 1.3

Revision History

Date	Version	Description	Author
22/October/18	1.0	First draft	
10/December/18	1.1	Update to phase plan	Antonio Rodriguez Esquire
10/December/18	1.2	Releases - Iteration C1 and C2 added	Antonio Rodriguez Esquire
10/February/2019	1.3	Edit to 1.4 References drafted	Dylan Perez
29/April/2019	1.3	Final	Antonio Rodriguez Esquire, Dylan Perez, Geordie Jones, Hunter Merritt, Sabyasachi Sahoo

Table of Contents

Software Development Plan

1. Introduction

1.1 Purpose

The purpose of the *Software Development Plan* is to gather all information necessary to control the project. It describes the approach to the development of the software and is the top-level plan generated and used by project managers to direct the development effort.

The following people use the *Software Development Plan*:

- The **project manager** uses it to plan the project schedule and resource needs, and to track progress against the schedule.
- **Project team members** use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.

1.2 Scope

This *Software Development Plan* describes the overall plan to be used by the <project name> project, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans document. The plans as outlined in this document are based upon the product requirements as defined in the *Vision Document* and Use Cases.

1.3 Definitions, Acronyms, and Abbreviations

See the Project Glossary.

1.4 References

For the *Software Development Plan*, the list of referenced artifacts includes:

- Vision Document
- System Requirements Specification
- Glossary

1.5 Overview

This *Software Development Plan* contains the following information:

Project Overview — provides a description of the project's purpose, scope, and objectives. It also defines the deliverables that the project is expected to deliver.

Project Organization — describes the organizational structure of the project team.

Management Process — explains the estimated cost and schedule, defines the major phases and milestones for the project, and describes how the project will be monitored.

Applicable Plans and Guidelines — provides an overview of the software development process, including methods, tools and techniques to be followed.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

2.2 Assumptions and Constraints

2.2.1 Budget Assumptions

We assume that the project will be implemented with our free labor and on completion will be maintained by Dr. Ma and his grant from the NSF.

2.2.2 Staffing Assumptions

Currently we assume that our team will be the only staff on the project however at its completion it will be maintained by Dr. Ma. This assumptions leads us to the constraint that the site will need to be easy to maintain and possibly implementing a custom cms for ease of use.

2.2.3 Equipment Assumptions

We assume that we all have our own workstations or access to one and also we will be utilizing a server on AWS to host the application.

2.2.4 Schedule Assumptions

We assume that we will need to have the site finished by SRS so that we are ready to present and that Dr. Ma will be able to upload labs into the future on a variable timeline.

2.3 Project Deliverables

2.3.1 Deliverables are delivered towards the end of the iteration or as they are completed. Completed is defined as code is written, tested, and reviewed by another team member.
2.3.2 If a deliverable is expected to be completed in an iteration but it won't be the team member assigned should let all parties know as soon as possible so that they may either assist or reschedule the deliverable to be due at a later date.
2.3.3 For list of deliverables and dates please see Project Schedule section.

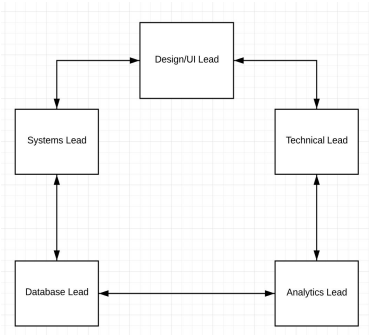
2.4 Evolution of the Software Development Plan

2.4.1

Version #	Changes	Date
0.0.1	First version of the SDP Plan.	10/22/18
1.3	Final version of the SDP Plan	4/29/19

3. Project Organization

3.1 Organizational Structure



3.2 External Interfaces

Google Analytics - used for collecting website traffic.

Sendgrid - used to send out emails to users and admins.

3.3 Roles and Responsibilities

The following table defines the roles and responsibilities for each team member.

Person	**Rational Unified Process Role**
Antonio Rodriguez Esquire - Design/UI Lead	Project Manager Design Reviewer User Interface Designer Technical Writer
Dylan Perez - Systems Lead	System Analyst Systems Architect Technical Writer
Hunter Merritt - Technical Lead	Integration User Interface/Design Technical Writer
Sabyasachi Sahoo - Software Engineer	Analytics Testing Technical Writer
Geordie Jones - Software Engineer	Database Design Testing Technical Writer

Anyone on the project can perform Any Role activities.

4. Management Process

4.1 Project Estimates

Project estimates is currently an assumption based on current requirements.

4.2 Project Plan

4.2.1 Phase Plan

A Work Breakdown Structure is being prepared, and will be provided in the next version of this document.

The development of the Eureka Labs System will be conducted using a phased approach where multiple iterations occur within a phase. The phases and the relative timeline is shown in the table below:

Phase	**No. of Iterations**	**Start**	**End**
Inception Phase	1	September 24th	October 10th
Elaboration Phase	1	October 11th	November 7th
Construction Phase	3	November 8th	April 21st
Transition Phase	2	April 22nd	May 1st

Phase	**Description**	**Milestone**
Inception Phase	The Inception Phase will develop the overall product vision, system requirements, and identify the stakeholders of the system. Use cases will be developed and prioritized along with high level Software Development Plan. At the end of the Inception Phase Professor Liran Ma will decide whether the planned system requirements have been met.	Requirements Review Milestone at the end of the phase Professor Ma will approve the different requirements for the system.
Elaboration Phase	The Elaboration Phase will establish architecture and design for the bulk of development effort in the next phase. During this phase we will produce a schedule of use cases for first and second Release. This phase will also identify high risk use cases for the second Release.	The Architectural Prototype Milestone marks the end of the Elaboration Phase. Milestone will be achieved when Professor Ma approves of the prototype.

Construction Phase	During the Construction Phase, remaining use case will be designed and analyzed. This phase will iteratively develop use cases until an operational version of the system is ready to be released. Testing and development will continue until the fully operational BETA is ready for the second release.	The R2.0 Release Milestone marks the end of the Construction Phase. At this point, the system is ready for deployment in the
Transition Phase	The Transition Phase will verify that all requirements have been met. All supporting document and training will transferred to the system owner.	The R2.0 Release Milestone marks the end of the Transition phase. All requirements described in the vision document have been met.

4.2.2 Iteration Objectives

High Level Objectives

- Address risk and any issues that come up during planning and development.
- Develop basic functionality and get approval from stakeholders
- Deploy application into an user friendly environment for the stakeholder
- Deliver technical documentation

The following table describes the iterations along with associated milestones and addresses risk.

Phase	**Iteration**	Description	Associated Milestone	Risk Addressed
Inception Phase	Preliminary Iteration	Define requirements, Software development plan, and vision document.	Requirements Review Milestone	Determine the risk associated with each requirement.
Elaboration Phase	E1 Iteration	Complete R1 use cases. Complete R2 high risk use cases	Architectural Prototype Milestone	Prototyping for the system owner. Risk identified and addressed.
Construction Phase	C1 Iteration	Implement and test R1.0 use cases. Deploy R1.0 Beta	R1.0 Beta Milestone	User feedback from Beta

	C2 Iteration	Implement and test remaining use cases from R1.	R1.0 Software Milestone	R1.0 reviewed and feedback received
	C3 Iteration	Apply patches for defects in software R1.0 Complete R2.0 Use cases	R2.0 Software Milestone	Addressing defect and issues from R1.0
Transition Phase	T1 Iteration	Deploy and test R1.0 in production environment	R1.0 Release Milestone	Continued testing to ensure packages are not missing.
	T2 Iteration	Deploy and test R1.0 in production environment	R2.0 Release Milestone	Continued testing to ensure packages are not missing.

4.2.3 Releases

Beta R1.0 - C1 Iteration

For our first construction phase iteration we selected 7 use cases and planned to complete all of them within 3 weeks. Along with the initial setup, we were able to complete 5 use cases and demo all 5 to the stakeholder.

UC002 is 75% complete and had additional changes requested from the demo. UC002 will move to the next construction phase iteration C2.

UC009 was not completed due to the complexity of the other lab use cases. UC009 will move to the next construction phase iteration C2.

Docker was complete, but because of the restructure of the Flask application the docker must be reconfigured.

- Use Cases & Initial Setup
    - Platform and Software
        - AWS setup of EC2 instance
        - MongoDB database
        - Flask Application
    - UC001 Content Creator / Educator request account
        - registration page
        - create user schema
        - adding users to database
        - create form class
        - form validation
    - UC003 Content Creator / Educator logs into the system
        - login page
        - create form class
        - validate user and password

        - create session role
    - UC004 System Administrator logs into the system
        - add roles to the user schema
        - admin page
        - View all users
    - UC008 Content Creator creates lab
        - lab creation page
        - create lab schema
        - create S3 storage buckets
        - upload file to S3
        - add S3 link to lab database
    - UC010 Content Creator deletes lab
        - delete action
        - remove from database
        - remove from S3

- Incomplete Use Cases
    - UC009 Content Creator edits lab
        - create edit page
        - update action
        - add action to dashboard view
        - update database
        - update S3
    - UC002 System Admin approves account
        - create pending user page (complete)
        - create edit profile action (complete)
        - update profile / approve account (complete)
        - create new user with admin role
        - remove admin role from selection
    - Docker setup
        - create nginx docker file

R1.0 - C2 Iteration

For our second iteration, we have selected 2 use case in addition to the use cases carried over. Continued testing and the implementation of automated test cases will also occur during this iteration.

- Completed Use Cases
    - Docker
        - configure file
    - Python automated test case
        - login test
        - register
        - add lab
    - UC002 System Admin approves account
        - create pending user page (complete)
        - create edit profile action (complete)
        - update profile / approve account (complete)
        - create new user with admin role
        - remove admin role from selection

    - UC005 User searches for lab topics
    - UC009 Content Creator edits lab
        - create edit page
        - update action
        - add action to dashboard view
        - update database
        - update S3
    - UC013 User rates lab

R1.0 - C3 Iteration

In this iteration, we did four use cases.

- Completes Use Cases
    - UC007 - User looks at the top labs
        - top labs page
    - UC006 - User post a question on the lab discussion
        - post question on lab
    - UC016 - System Administrator filters list of users accounts
        - User administration page
    - UC018 - System notifies user when account is approved
        - Email notifications

R2.0 - C4 Iteration

- Complete Use Cases
    - UC012 - Content Creator views lab analytics
        - Analytics page
    - UC014 - View analytics for all labs
        - Admin views analytics
    - UC017 - User request password reset
        - Email notification

4.2.4 Project Schedule

Iterations	*Use Cases*	*Dates*
Iteration 1	UC001 Content Creator / Educator request account UC003 Content Creator / Educator logs into the system UC004 System Administrator logs into the system UC008 Content Creator creates lab UC010 Content Creator deletes lab	November 8th - November 29th

Iteration 2	UC002 System Admin approves account UC005 User searches for lab topics UC009 Content Creator edits la	November 30th - December 21st
Iteration 3	UC007 - User looks at the top labs UC006 - User post a question on the lab discussion UC016 - System Administrator filters list of users accounts UC018 - System notifies user when account is approved	January 15th - February 5th
Iteration 4	UC012 - Content Creator views lab analytics UC014 - View analytics for all labs UC017 - User request password reset	February 6th - February 27th
Iteration 5	Testing and debugging	February 28th - March 21st
Iteration 6	Deployment and transition	March 22nd - April 18th

4.2.5 Project Resourcing

The number of staff required for this project is 5.

Each member will have to learn how to use Flask and Python for full stack development. In addition, each team member must familiarize themselves with Amazon Web Services.

4.3 Project Monitoring and Control

Requirements Management

The requirements for this system are captured in the Vision document. Requested changes to requirements are captured in Change Requests, and are approved as part of the Configuration Management process.

Schedule and Budget Control

Expenses are monitored by the project manager, and reported and assessed monthly. (See Reporting and Measurement below).

The project manager maintains a schedule showing the expected date of each milestone. The line items in the schedule include work packages assigned to individuals. Each individual who is assigned a work package provides %completion information to the project manager on a weekly basis. Changes in the schedule will be escalated to the project sponsors, who will then decide whether to alter scope in order to preserve target completion dates.

Quality Control

Defects will be recorded and tracked as Change Requests, and defect metrics will be gathered (see Reporting and Measurement below).

All deliverables are required to go through the appropriate review process, as described in the Development Case. The review is required to ensure that each deliverable is of acceptable quality, using guidelines described in the RUP for Small Projects review guidelines and checklists.

Any defects found during review which are not corrected prior to releasing for integration must be captured as Change Requests so that they are not forgotten.

Reporting and Measurement

Updated cost and schedule estimates, and metrics summary reports, will be generated at the end of each iteration.

The Minimal Set of Metrics, as described in the RUP Guidelines: Metrics, will be gathered on a weekly basis. These include:

Earned value for completed tasks. This is used to re-estimate the schedule and budget for the remainder of the project, and/or to identify need for scope changes.

Total defects open and closed – shown as a trend graph. This is used to help estimate the effort remaining to correct defects.

Acceptance test cases passing – shown as a trend graph. This is used to demonstrate progress to stakeholders.

In addition, overall costs will be monitored against the project budget.

Risk Management

Risks will be identified in Inception Phase using the steps identified in the RUP for Small Projects activity "Identify and Assess Risks". Project risk is evaluated at least once per iteration and documented in this table. The risks of the greatest magnitude are listed first in the table.

Ranking (High, Medium, Low)	Risk Description and Impact	on Strategy and/or Contingency Plan
High	Losing a member of the team	Distribute workload evenly
High	Server security breach	Follow best practices and validation testing

Configuration Management

Appropriate tools will be selected which provide a database of Change Requests and a controlled versioned repository of project artifacts.

All source code, test scripts, and data files are included in baselines. Documentation related to the source code is also included in the baseline, such as design documentation. All customer deliverable artifacts are included in the final baseline of the iteration, including executables.

The Change Requests are reviewed and approved by one member of the project, the Change Control Manager role.

Full backups are performed monthly and increments are performed nightly.

5. Annexes

The project will follow the RUP for Small Projects process, as tailored by the project Development Case.

Other applicable process plans are listed in the references section, including Programming Guidelines.