

Pager Project Development Approach

Abstract

This document describes the environment, the methods/practices and the tools used in the development of the pager. It covers how the team will operate, the artifacts, the expected standards and level of rigor required.

Version History

Date	Version	Author	Description
26/03/2020	1.0	Abyel Tesfay	First draft

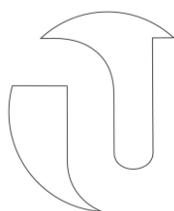


Table of Contents

1	Introduction	3
1.1	Document Purpose	3
1.2	Document Scope	3
1.3	Document Overview	3
2	Process Objectives	4
3	Summary of Approach	4
3.1	Interfaces	4
3.2	Standards	4
3.3	Alternatives Considered	4
4	Process Rigor	4
5	Selection of Practices	4
6	Selection of Tools	5
7	Improvement Strategies	5
8	Evaluation Criteria	5

1. Introduction

1.1 Document Purpose

The purpose with the following document is to describe the environment and designs the product will be developed with as well as the methods, praxis and tools used for the development. The document:

- Describes how the project group will work during development.
- Summarizes the praxis, tools, help and guides used.
- Indicates the level of rigor required for internal and external deliverables of product.
- Explains the procedures adopted by the group to control and maintain the development of the product.

1.2 Document Scope

The document includes:

- The environment of the software development in the project
- An approach to the development of the product and its purpose
- Praxis and tools used during the development

The document does not include:

- Project and program management

1.3 Document Overview

This document contains the following sections:

- Objectives – the goal of the selected approach
- Summary of Approach – high level view of the overall approach and the interfaces and standards constraining the approach.
- Rigor – The levels of formalization that are needed and what they apply to.
- Selection of Practices – which practices are being used and why.
- Selection of Tools - Lists tools to support the approach and how they support it.
- Improvement Strategies – how the approach will be tailored and improved over the lifetime of the project.
- Evaluation Criteria – describes how the results and effectiveness of the approach will be measured.

2 Goal

The purpose with the development plan is to let each member work with deliverables associated with technical areas they excel in, such as *web development* or *software programming*. By focusing on developing all deliverables of the final product in early stages of the project, The plan minimizes the risk of encountering errors and bugs later in the project, when all deliverables (the *pager-device* and the *web application*) are linked together. Through this method the group hopes to develop a product with good quality and durability. The plan also gives time for the project group to write their individual reports. The strengths and weaknesses of the development plan is listed below:

- + Discovering bugs and dependencies when connecting all deliverables early and solving them.
- + Members producing better result in their respective areas of expertise.
- More responsibility and tasks for each member.
- The project is very dependent in that all deliverables of the product are finished early to mid-project.

3 Summary of Approach

In the development plan, the project group is divided into two smaller *development* groups, where both groups are responsible for the production, maintenance and documentation of respective parts of the full *pager* product. The group members are divided into the following sub-groups:

Group 1	Abyel T, Adam L, Alexander T	Responsible for development and production of a <i>web application</i> , that communicates with <i>pager</i> -devices through a <i>remote backend</i> in cloud-service.
Group 2	Elias J, Mikael A	Responsible for development and production of <i>pager</i> devices. The <i>pager</i> should fetch messages from a remote backend in cloud-service, through available wireless networks.

During development of the deliverables, the groups will keep each other updated through daily scrum-meetings, where progresses and problems in the development are discussed. Through this both groups keep track of the ongoing development and understand the results that are expected by respective deliverables.

When all deliverables of the product are produced, the development groups will merge into the original project group. At this stage all members work together to meet the remaining **must-requirements** and given the available time and resources, fulfill **non-functional wishes** stated in the *requirements specification*. One important must-requirement at this stage, is the establishment of a connection between the *pager*-device and the *web application*.

3.1 Standards

There are no standards of importance that the project group needs to follow or consider. This is due to the small scale of the project and most members being inexperienced with projects.

3.2 Alternatives Considered

An alternative development plan is that **all** group members works together to produce every deliverable of the *pager* product in each sprint, with the purpose to deliver ready deliverables without any or few bugs. While the project group enjoys the benefit of having well made parts with few or none bugs and errors for building the final product, the group runs the risk of encountering bugs/errors related to the *connection between the web application and the pager device* later in the project. Therefore this approach was not adopted.

4 Rigor

The product in development is broken down into the following deliverables, where each is handled by one of the two development groups mentioned in section 3. Each deliverable is reviewed according to their function or type in the list below.

Deliverable	Approach to review deliverable
Interactive web application	All group members review/test the application.
The web application is connected to a remote backend and can exchange messages	Tester examines the connection and data exchanged between the web application and the remote backend through a series of tests.
<i>pager</i> device that can display received messages	Tester checks that the <i>pager</i> displays received messages correct through display.
<i>pager</i> can connect to wireless networks	The developers and the tester conduct a series of tests to check the connection.
<i>pager</i> can fetch messages from a remote backend provided by the cloud-service through network	Tester examines the functionality through tests and by letting the <i>pager</i> fetch pre-defined messages from the remote backend.

5 Selection of Practices

The following practices have been adopted in the development process and by group members.

- The source code for the *pager* is written in C, while the *web application* is written in JavaScript using React JS Library. Developers responsible for the web application chose React due to experiences with the library, C was also chosen by developers of the *pager*-device for the same reason.
- The project will make use of two separate repositories, one for the source code of the *pager* and another for the script/html files that will become the web application. This is done so that work on both deliverables can be separated.
- Members working with a *task* should always work on branches with the same name as the task. This is done so that new features and functionality can be developed without affecting the *master* branch containing working code from earlier iterations.
- Each member should *commit* and *push* changes or progresses made on their task to their repository regularly. This is so that all developers are updated with the latest changes at all times.
- When a *story* or large task is completed, the developer sends a *pull request* to merge their branch with the *master* branch. Members can then review the

additions or changes made. Additional changes can be made until the request is accepted and the branch is merged into master, marking the completion of said *task*.

- Members of the project group are recommended to use the “*lean coding*” method, in which a task is divided into several technical parts that should be completed within a time limit (10-15 min). Through this method, one can quickly identify if a certain part is trivial or something that needs to be solved through help or by identifying additional problems/tasks.

6 Selection of Tools

The following tools below will be used for the duration of the project

- Git will be used for version control and Github as hosting service for source codes, Github will also be used for *continuous delivery* and *continuous integration*
 - for github: <https://guides.github.com/>
 - for git: <https://guides.github.com/introduction/git-handbook/>
 - github actions: <https://help.github.com/en/actions>
- Google firebase, in which the web application will make use for hosting, authentication and database services
 - firebase: <https://firebase.google.com/docs/web/setup?authuser=0>
- Node.js, the JavaScript runtime environment that was used for developing and testing the web application.
 - node.js: <https://nodejs.org/en/>
- Node package manager (npm), a package manager for JavaScript, used for managing packets.
 - <https://www.npmjs.com/>
- stm32CubeIDE, an integrated development environment, specifically created for STM 32 microcontrollers/microprocessors
 - <https://www.st.com/en/development-tools/stm32cubeide.html>

7 Improvement Strategies

For improvement strategies, the group uses the sprint retrospectives, held after each completed sprint, for reviewing work and team performance as well as configuring the praxis and tools used for the coming sprints. During such retrospective, the group discusses the pros/cons and the effect the used tools and praxis had on the development. Based on the review, the group then selects those praxis and tools that are kept or removed or adds new choices onto the project.

8 Evaluation Criteria

For evaluation, the burn-down chart plays a vital role for asserting whenever the project group is making progress or falling behind schedule, in which measures can be taken. The measures can vary from increasing hours put into the project development to cutting non-functional features which may need additional work.