Configuration Management Plan

Advisely, An Education Support Application

Big Muscle Boys

Sam Elliott, Jens Hansen, Connor Langlois, Eli Legere, Eric Schessler, Spencer Ward

COS 420 UMaine Spring 2019

Professor: Sepideh Ghanavati

Table of Contents

1. Introduction

The purpose of this document is to outline how the software product Advisely will be maintained after initial release.

1.1 Background

Advisely is a scheduling support application designed to aid students. Advisely lets users generate schedules for their current and future college semesters. Included in Advisely are course consistency checks, graduation requirement verification, conflict avoidance, among other support functionalities. These schedules will be kept persistent with the user's account.

1.2 Life Cycle

As the usage of our app is heavily defined by college enrollment periods, our lifecycle will be centered about these periods. Thus, every year we will have two major releases which will include new functionality, bug fixes, and significant changes to the features the application currently contains. These releases will occur in the months of March and September. As well, between these two updates will come 1-2 minor updates. These updates will include minor alterations and bug fixes, but no new features.

1.3 Support Systems

The systems that support Advisely are React v16.8, Firebase v5.9.3.

1.4 Limitations

The primary limitations placed on the configuration management of Advisely are development time and information deficit. Time will be limited by team member availability past the end of the semester. Additionally, since course information can be unpredictable and subject to change, our application's information will always be imprecise to some degree.

2. Activities

2.1 Configuration Identification

The Advisely team will consider configuration items to items that need to be modified over time. By these conditions, the database and the user interface will be the only configuration items.

2.2 Configuration Control

2.2.1 Requesting Changes

Users may request changes through the Advisely support email address, advisely-support@googlegroups.com.

2.2.2 Evaluating Change

Scrum meetings will operate on a two week cycle, where every second meeting will be extended by up to 3 hours. During this time, the development team will evaluate the merit of each change request, starting with the oldest email that has yet to be evaluated. During this evaluation, an approximate time requirement will be assigned to each change. These evaluations will be used for planning future updates.

2.2.3 Approving or Disapproving Changes

During each meeting, a voting process, based on majority vote, will occur between the developers and scrum master to determine which changes submitted by users will be accepted. If a given change is not accepted, the corresponding change request will be archived for future reference.

2.2.4 Implementing Changes

Changes in the backlog will be divided into major and minor releases based on the size of the request. If a change is marked as major, it will be designed, implemented, tested, and released in the next major version. Otherwise, it will be developed in the next minor version.

2.3 Subcontractor Control

As this application is built upon React and Firebase, periodic modifications will be needed in order to implement the selected changes. In the case that a modification to either service negatively impacts the capabilities of Advisely, all other development will be paused until this issue is resolved.

2.4 Release Management Control

The software will be automatically built through continuous integration as the code is pushed to the repository. It will be released at a date determined before software development has begun. Additionally, code freezes should be put in effect two weeks prior to release to allow ample time for testing.

3. Management

3.1 Organization

The software is managed through the context of scrum meetings, where product owner and scrum master provide supervision and oversight towards compiling the previous sprint and organizing upcoming sprints. Organization of technology is primarily maintained by developers. Each is provided independent oversight of specific components, along with detailed tasks and duties each sprint.

3.2 Responsibilities

Title	Responsibility
Scrum Master	Assure that all other workers are upholding their responsibilities, as well as completing tasks as a developer.
Product Owner	Assure that the product goals are being met to sufficient quality in a timely manner.
Developer	Responsible for implementing user stories set forth each scrum and making changes described by QA analysts.
Quality Assurance Analyst	Analyze code and documentation produced by developers with the intent of discovering errors, bugs, or inaccuracies.

3.3 Directives

In the event that an update to Firebase or React negatively impacts the services offered by Advisely, all ongoing development will be halted until full functionality is restored.

4. Schedule

4.1 Release Sequence

Major updates to Advisely will be pushed twice per year, within the 6 weeks preceding the standard college enrollment period. Each of these updates will include major new features, updated course offerings, and fixes to application errors.

Between major updates, between one and two minor updates will be released. These will include fixes to significant bugs and minor ease-of-use changes. These updates will be released every 1 - 3 months, depending on the level of urgency.

4.2 Release Relationship

The development of a minor release will run parallel to the beginning stages of development to the next major release, as its primary function is to implement fixes for the prior release. A second minor release will attempt to supplement these fixes. Depending on its development lifecycle, this release may be merged into the next major release.

Major releases are scheduled based on enrollment deadlines. While minor releases are subject to mergers and shifts, major releases must be developed with a concrete and permanent deadline.

4.3 Release Schedule

The following is a sample of a typical release cycle:

Month	January	March	May	July	September	November
Release Type	Minor	Major	Minor	Minor	Major	Minor

5. Plan Resources

5.1 Infrastructure

The defined software requires a Firebase backend and React frontend as infrastructure.

5.1.1 Functionality

The Firebase backend provides the necessary functionality for authenticating users and storing information. It does so through its proprietary authentication protocol and Firestore database.

The React frontend provides the ability to display information, such as courses and schedules, and to accept user input for the creation of user-defined schedules.

5.1.2 Performance

All activity within the Firebase backend is monitored by Firebase's servers to provide real-time scaling as required. If there is a large number of users on the platform, the backend will scale to support them.

Similarly, the React frontend prevents bottlenecks by hot-swapping components as they are needed. If the user is currently not viewing a particular component, its memory will be deallocated and made available for future allocation.

5.1.3 Security

No information gathered and stored by Advisely is safety critical. Further, all information gathered is stored within a Firestore database, so safety concerns should be directed at their infrastructure. Finally, authentication is handled by Firebase. Advisely will not store any user email addresses or passwords, only tokens that correspond to accounts. As before, concerns about the security of email addresses and passwords should be directed to Firebase.

5.1.4 Availability

As described above, in the event of app failure for any reason, the weight of our development team's efforts will be directed to restoring the app. Hence, although our app will not always be available, it should be available at least 95% of the time.

5.1.5 Space Requirements

The backend Firestore database shall only need a maximum of 10 GB of data. This includes all user-generated and course data due to the small size of user and course data gathered.

5.1.6 Costs

For testing purposes, Advisely uses the free tier of the Firebase subscription. That means it will remain free to host for up to 100 simultaneous connections. If the app needs more connections, Advisely will have to upgrade to the paid tier of Firebase ($25 per month).

5.2 Training

All developers need sufficient expertise with React in order to work on Advisely. To ensure this level of expertise, developers need to complete the following tutorial. This tutorial can similarly be found on reactjs.org under their tutorial section.

6. Plan Maintenance

As Advisely will have two major updates per year, our plan maintenance schedule will mirror this. After each major release, the current scrum master will dedicate the following sprint to evaluating each aspect of the CMP. Should they find any aspects which are unnecessary, redundant, ineffectual, or harmful to the development of the app, they shall be brought to the attention of all members of the development team. At this point, a vote will be performed to verify these changes. Should any individual vote pass, the necessary changes shall be made to CMP by the following scrum meeting. In the event of a tie, the scrum master's vote shall be weighed more heavily and break the tie.

https://www.fsa.usda.gov/Assets/USDA-FSA-Public/usdafiles/SDLC-non-secure/FSA%20Systems%20CMP%20JW%20sig%207-11-2016%20GP%20sig%207-12-2016.pdf