Climate Visualization For Natural Resources

Test and Validation Plans

Center for Sustaining Agriculture and Natural Resources

Mentor

Dr. Kirti Rajagopalan

Gemini

Ryan Torelli, Matt Bourland, Roosevelt Young, Hasnain Mazhar

CptS 423 Software Design Project II
Aaron S. Crandall, PhD

TABLE OF CONTENTS

I. Introduction

I.1. Project Overview

The Center for Sustaining Agriculture and Natural Resources (CSANR) [1] has requested a web-based tool for aiding sustainable practices of agriculture in the face of climate change. The primary user group is agricultural professionals of the United States who advise agricultural producers on crop and livestock selection, growing and rearing practices, and pest management. In 2016, a design team delivered one feature toward development of the decision tool. During the 2017-18 academic year, a second design team is undertaking development of an additional feature.

This design project aims to build a feature that shows through data visualization how climate change impacts land used for cattle production. The feature comprises a map of the United States and a series of graphs. The map visualizes data on a large spatial scale and the graphs visualize data on a smaller, aggregated spatial scale.

The main functionality of the application feature for testing is the drop-down menu, splash page, and data view page.

I.2. Test Objectives and Schedule

The main objective in our testing plan is to have system code pass all of the testing activities detailed below. Some of these tests focus on the code, such as unit testing and integration testing. Other tests focus on the functionality and behavior of the code, such as system testing, performance testing, and user acceptance testing. The resources required to complete these activities are access to our code base as well as the RStudio integrated development environment. The testing activities will be conducted in order from smallest to largest units, followed by functional testing. The products that will be delivered for testing are the driver files for unit and integration testing and brief write-ups for other testing activities.

I.3. Scope

The purpose of this document is to outline our strategy and methods for testing our system.

II. Testing Strategy

We followed the strategy outlined below:

1. Document the test case configuration, test data, and expected results.
2. Create unit test driver and conduct unit tests. Modify as needed until all tests pass.
3. Create integration test driver and conduct integration tests. Modify as needed until all tests pass.
4. Conduct functional testing based on functional requirements.
5. Determine passing values for performance testing.
6. Conduct performance testing based on passing values.
7. Conduct user acceptance testing, including user testing.
8. Submit all test documents and reports.

III. Test Plans

III.1. Completed Testing

III.1.1. Functional testing

The goal of functional testing was to identify the most relevant use cases and testing those with the highest probability of failure. We tested menu options for each boundary type - state, county, or district - and selected at random five bounded regions. Selection of a bounded region generates a plot corresponding to all forecast data aggregated for that region. We verified that both the aggregated data and plot matched the data selected.

III.1.2. User Acceptance Testing

We tested for user acceptance by confirming that the application meets every requirement specified by the client. We allowed the client to use the application and provide feedback for revision. We maintained a checklist of client requirements (Appendix) that we used to test item by item. After the application passed the checklist, we notified the client, made the application available, and scheduled a meeting for discussion of feedback. We incorporated feedback as a set of final revisions and delivered the application with documentation and a user help video to a predetermined web address.

III.2. Should be Tested

III.2.1. Unit Testing

To conduct unit testing on the system, we will start by breaking down the code into the smallest units possible. These units are individual functions. We will write test drivers to call each function and compare test output to expected output. If they match, the test passes. If a test fails, we will modify the function or use a different approach to obtain the desired functional outcome. Completing unit testing will increase our level of confidence in the functionality of the system. The test driver will be an R file.

III.2.2. Integration Testing

To conduct integration testing, we will take code units and test the interaction between them. We will test code units using as input the output of another unit. Similar to unit testing, we will check to make sure that the expected output matches the test output. If they match, the test passes. If a test fails, we will modify the function or use a different approach to obtain the output we need. Completing integration testing will increase our level of confidence in the functionality of our system. The test driver will be an R file.

III.2.3. Performance testing

The goal of performance testing is defining measures of performance and taking measurements under a variety of conditions. We will record and analyze the application’s response time in low, normal, and high stress situations. We will use a load test tool included in the Shiny library package, which records an application’s response time under different conditions of real-world use and server settings. Using the tool will help us record user interactions and number of concurrent users for a given duration, and playback results. If the results from performance testing are within bounds for performance requirements, we will have ensured client acceptance with respect to performance.

III.2.4. Usability testing

A major testing component incorporated in the checklist of client requirements is usability testing. The client has recruited a group of researchers and agricultural professionals to participate in testing a series of use cases. A protocol will be developed that standardizes three tasks. Each task will be described completely; questions will be added to each task to evaluate user understanding and task success. Some participants will take part in testing the web application remotely while others will be observed directly. The results of usability testing will be reviewed with the client for preparation of a set of initial revisions. The opportunity for client testing and feedback will follow the completion of revisions from usability testing.

IV. Environment Requirements

Environmental testing will include front- and back-end testing. The server for the application runs R and required library packages on a 64-bit CentOS Linux platform managed by the College of Agricultural, Human, and Natural Resource Sciences (CAHNRS). For the back-end, the primary concerns are memory leaks and exceeding server space allocation. In testing system requirements, we will monitor logs for system crashes and server space. For front-end testing, we will test function and aesthetics on different operating systems and web browsers. We will ensure coverage of the three most popular browsers, as determined by published sources, on Windows, Linux, and MacOS, as compatibility permits. In particular, our tests will focus on ensuring that such aspects as scaling and color are similar among combinations of operating system and web browser. It is paramount that the application have consistency of user experience so that all users may appreciate climate forecast visualization regardless of system and software.

V. Results

Everything that was tested works how we expect it to and the application runs flawlessly. The code coverage of our unit and integration testing cannot be determined because those tests have not been conducted yet. Confidence in the reliability of the system can be improved by completing the testing strategy including all of the tests in section III.2, “Should be Tested.”

VI. Glossary

Agriculture Professional - Crop consultant who advises on crop choices or university extension staff member that advises on best practices

CentOS Linux - Linux distribution sourced from Red Hat

Leaflet - Open-source JavaScript library for interactive maps

Shiny - Open-source R package that provides an elegant and powerful web framework for building web applications using R

R - Programming language for statistical computing and graphics

RStudio - Integrated Development Environment for R

VII. References

[1] CSANR. (2017). *Center for Sustaining Agriculture and Natural Resources* [Online]. Available: http://csanr.wsu.edu

Appendix

Functional Requirements

The application shall orient the user to region of interest with a map on the splash page, provide data categories to query on a drop-down menu, and display responses to queries on multiple data view pages.

Module: Splash page

Map: The splash page shall display a centered map of the mainland United States
- Source: The client requested that Rangelands display a map of the United States. Other features display a map and the client wants consistency among features.
- Priority: Level 1. This requirement is essential.

Zoom: The map shall zoom in and zoom out
- Source: The client requested that Rangelands support zoom like other features.
- Priority: Level 2. This requirement is desirable.

Data Points: The map shall display discrete data points as pins
- Source: The client requested that Rangelands display mapped data using a similar convention as other features.
- Priority: Level 2. This requirement is desirable.

Color-coded Data: The data shall be coded by color to indicate magnitude of values
- Source: The client requested that Rangelands code mapped data values by color.
- Priority: Level 2. This requirement is desirable.

Legend: The map shall have a legend
- Source: The client requested that the map have a legend to clearly express the data color scheme.
- Priority: Level 2. This requirement is desirable.

Module: Drop-down menu

Indicators: The drop-down menu shall include options for the user to select climate change indicators
- Source: The client requested that Rangelands have a drop-down option for the user to select indicator.
- Priority: Level 1. This requirement is essential.

Boundaries: The drop-down menu shall include an option to show county or congressional boundaries.
- Source: The client requested that Rangelands have a drop-down option for the user to select map boundary. It is important to some of our stakeholders (politicians) to be able to get information by county or congressional district.
- Priority: Level 2. This requirement is desirable.

Overlay: The drop-down menu shall include an option to change the overlay on the map (satellite/map/hybrid).
- Source: The client requested that Rangelands have a drop-down option for the user to select map overlay. Supporting different overlays is desirable in satisfying different users. For example, an agriculture professional might want to see the satellite view while a politician might want another view.
- Priority: Level 3. This requirement is extra.

Climate Models: The drop-down menu shall include an option to select from among multiple climate models
- Source: The client requested that Rangelands have a drop-down option for the user to select climate model. There are up to four different climate models. Some users might want to view one model and others several.
- Priority: Level 1. This requirement is essential.

Time Period: The drop-down menu shall include an option to modify the time period of concern for a climate model.
- Source: The client requested that Rangelands have a drop-down option for the user to select time period. The time period of concern is a significant source of information for the stakeholders.
- Priority: Level 2. This requirement is desirable.

Module: Data view page

Net Primary Production: The data view page shall display information on the net primary production climate change indicator.
- Source: The client requested that Rangelands display net primary production. This climate change indicator is one of the main pieces of information that our stakeholders seek.
- Priority: Level 1. This requirement is essential.

Interannual Forage Variability: The data view page shall display information on the interannual forage variability climate change indicator.
- Source: The client requested that Rangelands display forage variability. This climate change indicator is one of the main pieces of information that our stakeholders seek.
- Priority: Level 1. This requirement is essential.

Heat Stress: The data view page shall display information on the heat stress climate change indicator.
- Source: The client requested that Rangelands display physiologic heat stress on cattle. This climate change indicator is one of the main pieces of information that our stakeholders seek.
- Priority: Level 1. This requirement is essential.

Relative Fraction of Woody vs Herbaceous Plants: The data view page shall display information on the relative fraction of woody plants as compared to herbaceous plants climate change indicator.

- Source: The client requested that Rangelands display inedible to edible forage. This climate change indicator is one of the main pieces of information that our stakeholders seek.
- Priority: Level 1. This requirement is essential.

Aggregate: The data view page shall display information on an aggregate of all the climate change indicators.

- Source: The client requested that Rangelands display an average of all indicators. While some of our stakeholders may be concerned with only one or two climate change indicators, many will be concerned with all of them. The aggregate is a great tool to get a look at all of them together and get a feel for the big picture.
- Priority: Level 1. This requirement is essential.

Non-Functional Requirements

<u>Quality requirements</u>

Usability testing: The client requested testing of the application by a volunteer group of stakeholders.

User help: The client requested that the design team produce a user guide in the form of a video

Browser compatibility: The client requested that Rangelands be consistent among the current updates of the three most popular web browsers: Chrome, Safari, and Internet Explorer

Scaling: The client requested that visualizations scale to screen size

Resolution: The client requested that visualizations display without pixelation at 1280 x 720 resolution

Colors and font: The client requested that color schemes and font be consistent from page to page and most closely match other features

Indicator descriptions: The client requested that descriptions of climate indicators appear on data view pages

Performance: The client requested that load times of data view pages not exceed 30 seconds

Storage: The client has acknowledged that there is a tradeoff between performance and storage and requested that storage not exceed provided data size by more than a factor of three