Poseidon's Justice

Project:	What-If Scenarios for Water Rights
Team Members:	Michael Stickel
	Yosef Gamble
	Alex Romano
	Zayne Betts
	Erin Palmer
Faculty Advisor:	Dr. Lulofs
Client:	Dr. Pease
Date Due:	November 10, 2014

Revision Date	Change(s) Made	Name
November 7, 2014	Initial Revision	Michael Stickel
November 7, 2014	Introduction added	Michael Stickel
November 7, 2014	Project Overview added	Alex Romano
November 7, 2014	User Interface added	Alex Romano
November 7, 2014	Project Management added	Erin Palmer
November 7, 2014	Testing Process added	Erin Palmer
November 7, 2014	Title Page added	Michael Stickel
November 7, 2014	Table of Contents added	Michael Stickel
November 7, 2014	Feasibility added	Zayne Betts
November 7, 2014	Document standards added	Zayne Betts
November 8, 2014	Requirements added	Yosef Gamble
November 10, 2014	Intro to Project overview added	Alex Romano
November 10, 2014	Intro to UI added	Alex Romano
November 10, 2014	Conclusion added	Michael Stickel
November 10, 2014	Intro to Testing Process added and more content	Erin Palmer
November 10, 2014	Table for Risk Management added	Erin Palmer
November 10, 2014	Quality Assurance summarization added	Erin Palmer
November 10, 2014	Table of Figures added	Erin Palmer
November 10, 2014	Added caption to map	Alex Romano
November 10, 2014	Added our app design prototype	Alex Romano
November 10, 2014	Revised Requirements.	Yosef Gamble
November 10, 2014	Revised Functional Task	Yosef Gamble
November 10, 2014	Added Retrospective	Yosef Gamble

TABLE OF CONTENTS

Table of Tables?

INTRODUCTION

Central Washington University's Center for Spatial Information has requested features that will allow authorized users to create what-if scenarios for determining the best distribution ratio for water rights in Kittitas County. Our project will be building off of a web-based system that they currently have in place that utilizes ArcGIS for displaying mapping and monitoring water rights and water distribution in the Kittitas County.

The main feature that we will be implementing in our project is the ability for the User to determine different what-if scenarios for water rights in Kittitas County depending on the amount of rain fall at a given time. The User will be able to set theoretical levels of rain fall by use of a slider, prompting the interface to display the required data. To do this, the interface will be linked to both a climate database and a geo-spatial water database.

By providing this documentation, we hope to include those involved with this project throughout the course of our iterative process. Although we will continue to meet with the Client and our Project Advisor, this report will provide a formal means of monitoring the project's progress that can be referred to at any time as well as providing a guide for us in the creation of the project. The major sections that will be covered in this document include:

- Project Overview – provides an overview of the project
- Project management - explains how we will be tackling the issue
- Requirements – describes the requirements given to us by the client
- Design – provides a look into the design of our project
- Quality Assurance – insures a plan to produce a quality, uniform project
- Conclusion – provides an overview of the entire iteration and what needs to be worked on for iteration #2

PROJECT OVERVIEW

In this section we will identify all parties involved in this project and their roles. We will go over the application in its current state and what is to be added, and talk a little bit about we plan to solve this task.

Our client is Dr. Michael Pease, a professor in the Geography department of Central Washington University. He has requested that we expand on an already existing web application that he and other geographers in the Kittitas area use currently. Our stake holders involved are Dr. Michael Pease, our advisor, Dr. Ed Lulofs, Center for Spatial Information (CSI), and our team. What the application does now is it allows people to see a map of the Kittitas area and all the bodies of water contained in it. From the map view you are able to click on bodies of water to get more information about from the Department of Ecology.

What Dr. Pease would like to do with this application is to offer "what-if" scenarios to water levels in the area. These scenarios will test what would potentially happen if a certain body of water got more or less water than usual. Our solution is to add a toolbar to the existing application that will allow the user to test these scenarios. This solution needs to be developed rather than bought

because we are adding features and usability to an existing application. As far as we know there is nothing you can buy to do this for you.

PROJECT MANAGEMENT

PROCESS MODEL & TEAM ORGANIZATION

Our team will be using a combination of a modified waterfall variant, with verification and validation after each phase, and a few agile practices. Such agile practices include weekly standups (rather than daily), and bi-weekly "sprint" meetings. At first these sprints may not end in a functional iteration, but in the future they will be; these sprints are also the times when we plan to meet with the client in order to gather more requirements or verify that given requirements were fulfilled.

[handwritten: → do you mean end of sprint reviews?]

[handwritten: Pair programming? Estimation Technique? Roles?]

RISK MANAGEMENT

The project uses ArcGIS, a program for mapping geographical information, which no one in the group has experience with and poses a potential learning risk. Rather than have everyone become proficient with ArcGIS it would be better if everyone grasps a basic understanding of it, and have only one person who specializes in operating it. In addition, it has been stated by the client that the features initially wanted may not provide enough work for the project duration, but they also reassured us that there are "extra" features that they have thought of. Lastly, a risk that is present for every group, the lack of time to work on the project due to other classes, jobs, or other reasons. The best way for our group to combat this problem is to keep up good communication and plan accordingly around group members' free time.

[handwritten: My concern do! equh Where this?]

Risk	Likelihood	Impact	Mitigation Strategy
Team must learn ArcGIS	High	High	• Have all members review introductory material • Have key members review more in depth to ensure certain features work properly
Lack of time to commit towards project	Medium	High	• Good communication between group members • Re-prioritization of certain tasks to group members with more time
Not enough work given by client to keep group busy	Low	Medium	• Ask client about "extra" features they may want • Spend more time on unit testing, manual testing, and automation *[handwritten: ← not sufficient]*
A team member fails CS 480	Low	High	• Assist team members in their work and learning • Create progress reports/daily stand ups to track their progress.

TABLE 1: RISK MANAGEMENT ANALYSIS

[handwritten: Consider risk categories from SISP]

2

SOFTWARE DEVELOPMENT TOOLS

The project will involve coding in JavaScript and PHP, which most of the group is already familiar with. The main IDE being used to code will be NotePad++. Issues found will be tracked in a Jira database and assigned to the appropriate team member to fix. Also on Jira, we will be managing stories and breaking them down into subtasks and assigning them accordingly. For our version control we will be using GitHub. The database given to us is a variant of SQL and can be managed similarly to a regular SQL database.

[handwritten: really? I think you can find something better.]

[handwritten: This is a hosting service not VC (F## Git is the VC)]

[handwritten: not relevant?]

REQUIREMENTS

This project consist of three main requirement categories, User interface, Database, and Geospatial. The user interface requirements pertain to the web design languages that allow users to view and simulate data. Database requirements provide the information for the pop-up menus when a user clicks on a body of water on the map, as well as the add-on feature of simulating water rights based on scenarios. The third requirement category, is geospatial, which allows a user to visualize and interact with data provided through the database.

If the three categories of requirements are satisfied the program should solve the problem easily and efficiently.

FUNCTIONAL REQUIREMENTS

The additions to the user interface rely on the extended use of HTML, JavaScript, and PHP to interact with the existing PostgreSQL Database. The database pulls geospatial information from various sources and merged with ArcGIS Desktop.

The software required for this program is PHP in order to allow the user interface to interact with the database. PostgreSQL is used to import and organize geo-spatial information, and allows for ArcGIS information to interact with real-time information of bodies of water in Kittitas County. ArcGIS is required to compile information provided by public databases about water levels and location geo-spatial information of water ways.
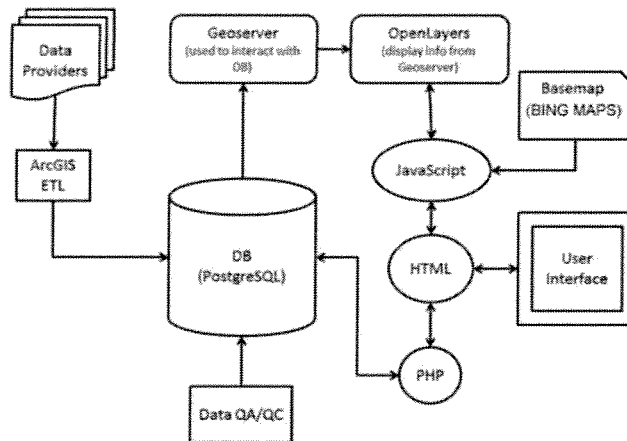
Figure 1 _Center caption_ → _Process Model_

The team is utilizing planning poker in order to estimate the time and organize the amount of tasks required to complete the program addition. Each team member is given a number of cards which contain a number of possible days may be spent working on a specific task.

First, we listed out the number of tasks. Then, we went through each task one by one to estimate how much time that each team member thought that each task would take.

As a team, we decided on X## of tasks, and the following story points and priority for each task:

FUNCTIONAL REQUIREMENTS TASKS

ID	Description	Priority	Story Points
1001	Create a user interface (slider, buttons for simulation)	High	1
1002	Connect user interface to database for real-time manipulation.	Medium	2
1003	Add predicted water levels to database	Low	2
1004	Create scenario outcomes for land owners in Kittitas county based on predicted water levels	High	5
1005	Design feature to compare predicted data with simulated data	Medium	4
1006	Implement Randomizing feature for simulation	Low	4
1007	Visualize resultant data	Low	2

caption?

not use case
Gannt

use cases?
uc diagram?

4

NONFUNCTIONAL REQUIREMENTS

The CSI program should give the user the ability have constant access and simulate of data for water levels in Kittitas County, and based on the simulation results, have results with 15-20 seconds.

The non-functional requirements includes run on all major browsers including Internet Explorer, Firefox, and Chrome. Additions to the program should be non-intrusive to current features on to the existing system, and should allow for the user to seamlessly manipulate and simulate water levels for Kittitas County based on historical and current geospatial data for Kittitas County. Minimum memory size to should not exceed 512 Megabytes, and should be accessible with a server uptime of 99% on an average day. If a visual component is added to the program, the trade off to memory would be to increase the minimum memory size to 1-2 Gigabytes.

FEASIBILITY

Completing the essential requirements for the system is very feasible in the time we are given. Because the project is building onto an existing system all of the requirements are additions to the current system in order to create more functionality. The essential additions to the system will allow users to run simulations to determine who has rights to the water that is available depending on availability. Other potential additions that could be made in the enhanced system could have to deal with the impact the water rights would have economically.

SYSTEM

ESSENTIAL SYSTEM

The essential system for the water right's simulator must allow for multiple scenarios, simulate water availability for 3-4 years, use real world predictable data in the simulations, and display the information using the existing system.

TABLE 2: BARE BONES VERSION

Id	Description
	Multiple scenarios
	Simulate availability
	Use predictable data
	Display using existing system

ENHANCED SYSTEM

The enhanced version of the water right's simulator will include all the features of the essential system, as well as features to predict economic impact.

TABLE 3: ENHANCED VERSION

| Id	Description
	Multiple scenarios
	Simulate availability
	Use predictable data
	Display using existing system
	Predict economic impact

DESIGN

The design for a majority of this project has already been created, so the extent of the design process is to add additional modules that will separate the simulation functionality from the current visual components of map layers that present reservoirs, rivers, wells, etc.

USER INTERFACE DESIGN

In this section we provide a look and description of what the user interface does currently. Then we give an idea of what the user interface should look like and do once we are done with this project.

As mentioned before much of the user interface of this application is already in place. We are adding features to the already existing application (pictured below). The bulk of the user interface is a map of the Kittitas area with emphasis on the surrounding bodies of water. To the right of the map there is a search feature to find bodies of water, and to the left you can find a drop down folder of optional layers you can add to the map.

FIGURE 1: CURRENT APPLICATION

6
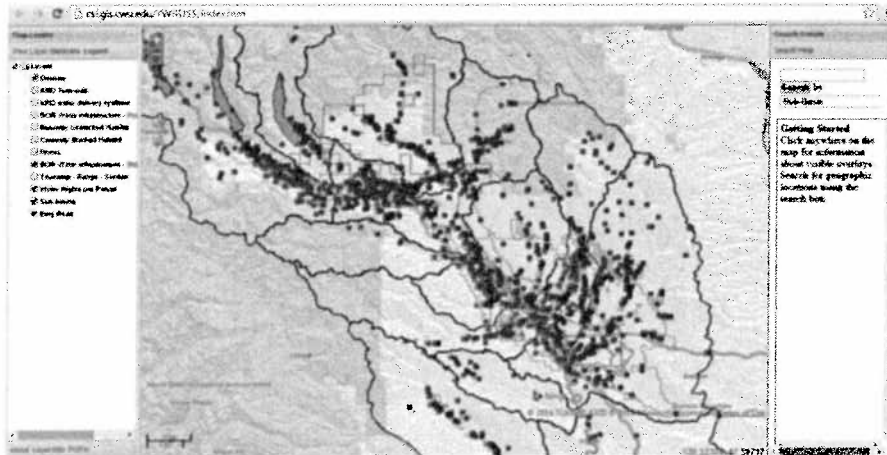
Our main requirement Dr. Pease has requested is a "what-if" scenario section. This section will be placed in the white space to the left under all the folder items. It will feature a slider that the user can adjust to how much water for a particular body of water (e.g. 20%, 100%, 200%, etc.). Beyond that there will be an option to change the climate schema, and an option to set the time this scenario will take place over. That portion will look something like what we have pictured below.

FIGURE 2: OUR ADDITION

QUALITY ASSURANCE

ignore *this*

In order to assure quality we will be using version and branch control to separate "stable" code and "unstable" code. This unstable code will be peer reviewed and then merged into the stable code. Furthermore, unit tests and manual tests will be done to ensure that the project functions as intended. Bi-weekly meetings will be held with the client as well to ensure that the requirements given have been fulfilled.

trunk

DOCUMENT STANDARDS

The standards we will try to uphold while writing our documentation consists of regular revision and addition, keep one voice, and use the desired format. Because we do not have a documentation lead in our team we will be evenly distributing sections of the documentation amongst the team members. We will spend parts of each team meeting discussing what needs to be added or updated in the documentation. Members who are most familiar with a specific portion of documentation will likely be chosen to work on that section. The team will select a date to hold a documentation-based meeting in order to revise and unify the documentation. Revisions to the documentation will attempt to make it as clear and consistent as possible and make it look professional and keep to the desired format.

TESTING PROCESS

Our goal for testing is to design and produce code before making tests. We would like the minimum for testing to at least implement unit tests and have automation be a last priority. This will catch most of the problems at the base level before they occur. Automation would catch everything else that gets past the base level, but time management might be a concern when implementing automation.

The project aims to have 90% code coverage in unit testing and will be using existing frameworks to carry out the unit tests. For the PHP unit tests PHPUnit plans to be used, and for the JavaScript unit tests QUnit will be the framework being used. These frameworks are simple to use and implement, while also providing the tools necessary to ensure quality code.

In addition to the unit testing, manual testing will also be part of the testing process, whether it to be to verify a bug that is filed can be reproduced, or whether it be to test new features. Jira will be a key instrument here as bug reports will dictate the paths for the manual testing (illustrating the bugs ran into by the client or bugs ran into while testing a new feature). The client will be manually testing our product versions and be keeping us up-to-date on new or on-going bugs via Jira or e-mails.

In the future, if time permitting, we hope to have automation for our project. This automation will implement a WebDriver framework, most likely Selenium WebDriver, and run the tests on Sauce Labs in order to cover multiple browsers and operating systems.

CONCLUSION

The problem presented to us by the client was to provide select users the ability to determine what if scenarios for water rights within Kittitas County. In order to solve this problem, we will be providing a means for the user to select the level of water fall at a given time by use of a slider that we will be adding to the clients web interface that they already have in place. Selecting this water fall will prompt the interface to access the database and display possibilities for water rights in Kittitas County.

RETROSPECTIVE

For this iteration, we spent a lot of time trying to get in contact with the client. We were able to get in contact with our advisor but due to class schedules, the entire team was not able to commit to being a part of every meeting. Unfortunately starting late has hindered progress to make the latest iteration of code to work. The server that hosts the project is frequently down, and the team has to rely on information presented in documents provided by the client to plan out the software features.

As a team, we established a time to meet for our weekly meetings and everyone was able to attend.

8

What can be improved is our communication with the client and establishing contact personally rather than via email or phone the first time around.

Our plan to address our communication with the client is to establish weekly meetings during their office hours to ensure that the team has a set time to be able to show how the product is being built so far and if requirements are being met along the way.

STATUS REPORT

This section provides a status report on the progress of this project, including user cases that are currently being worked on and a burn down chart for the predicted timeline of the project.

CASES

Project Status

TABLE 4 : Status

Use Case Id	Summary	
1	Moves Slider	In Progress
2	Chooses Simulation Years	In Progress
3	Choose Climate	In Progress
4	Choose Randomize Climate	Planned – Iteration #2
5	Submit	Planned – Iteration #2
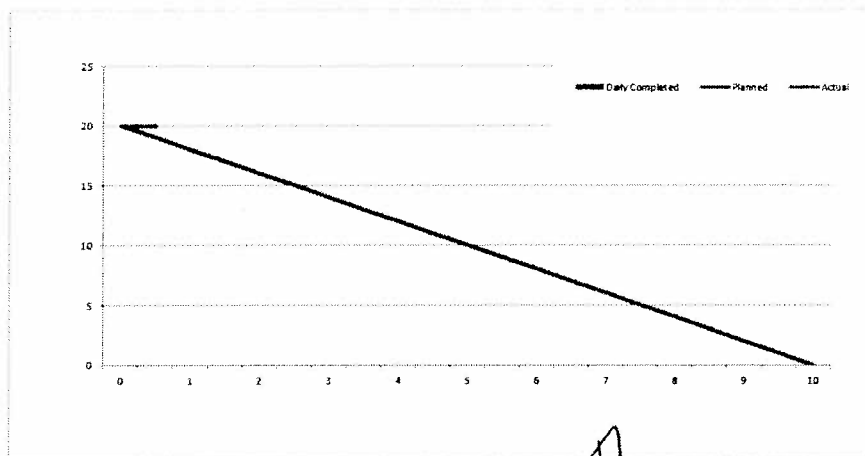
FIGURE 3: ITERATION 1 BURNDOWN *(EFFORT IS IN STORY POINTS)*

Only shows ideal, not actual for iteration

9

Project Metrics

TABLE 5

Metric	Result	
Bug Reports	Closed: 0	Open: 1
Unit Tests	Passed: 0	Failed: 0
Code Coverage	Statement: 0%	
Mutation Testing	Killed: 0%	
Static Analysis	Bugs Found: 0	

APPENDICES

TABLE OF FIGURES

After TOC

	Professional Quality (A-level)	Expected (B-level)	Acceptable (C-level)	Problems (D or F-level)
Front Matter (10 points)	• Title page contains team name, team logo, authors name(s), client name, advisor name and project name • Revision history is complete, up to date and demonstrates a cooperative work in an iterative manner. • Table of contents/figures/tables is complete and does not contain any undefined references	• One piece of information missing from title page • Revision history shows 'big bang' approach to creating the document • One undefined reference in a table of contents/figures/tables	• Two piece of information missing from title page • More than one undefined reference in a table of contents/figures/tables	• No title page • No revision history • No table of contents/figures/tables
Introduction (20 points)	• Briefly describe the real-world problem • Briefly mention the most important features and constraints of your program • Describe the purpose, scope, and intended audience of this document. Summarize and preview the major sections that follow.	• One item missing	• Two or more items missing	• No introduction provided
Project Overview (20 points)	• Identify the client, stake holders, and the intended users of your system • Explain background information on the general factors that affect the product and its requirements • Provide a complete description of the problem being solved • Include a justification for a computerized solution to the	• One item missing	• Two or more items missing	• No project overview provided

Criteria			
problem. Explain why a program needs to be developed rather than just bought • Describe the main features of the proposed system			
Project Management • Process Model & Team Organization • Risk Management • Software Development Tools (20 points) 14	• Explain what software process model the team will use. • If using an agile development methodology, what agile practices will you use? • Provide a risk management plan outlining the project risks and plans to address them • Software engineering tools for the project are described	• Process model description lacks details • Agile practices (if used) not detailed • Risk management plan not very through or only trivial items listed • Generic descriptions of the software engineering tools are provided, no specifics given G, 4/146 not VC	• No process model described • No agile practices described • No risk management plan • No software engineering tools described for the project	• No project management information provided
Requirements (40 points) 25	• Functional and non-functional requirements have all the characteristics of good requirements • Use cases used to describe requirements • All requirements are uniquely identified • All requirements are specified to be testable • All requirements have priorities	• High-level functional and non-functional requirements are missing a few good requirement characteristics • Most requirements are uniquely identified • Most requirements are specified to be testable • Most requirements have priorities • Most requirements have	• Use cases not used to describe requirements • Few requirements are uniquely identified • Few requirements are specified to be testable • Few requirements have priorities • Few requirements have story point estimates	• No requirements given

Category				
	• All requirements have story point estimates • Requirements are presented in both a table and a use case diagram(s)	• story point estimates • Requirements are presented in both a table or a use case diagram(s), but not both		
Feasibility (10 points) 8	• Feasibility analysis provides a complete high-level description of the essential and desired systems • All requirements are divided between essential and enhanced versions of the system	• Feasibility analysis describes two systems, but descriptions are very general • Most requirements are divided between essential and enhanced versions of the system	• Feasibility analysis attempted, but of poor quality • Only one system described • Few requirements are divided between essential and enhanced versions of the system	• No feasibility analysis provided
User Interface Design (20 points) 20	• Text description completely describes the user interface • Mockups of the user interface are presented • Description of the user interface is tied to all of the use cases	• Only a textual description of the user interface is provided. • Description of the user interface is tied to most of the use cases	• Only a textual description of the user interface is provided. • Description of the user interface is tied to a few of the use cases	• No user interface design provided
Document Standards (10 points) 10	• Standards and procedures for quality documentation described **and** the document shows adherence to standards and procedures.	• Standards and procedures for quality documentation described but the document shows evidence of minor deviations from the standards and procedures.	• Standards and procedures for quality documentation described but the document shows evidence of major deviations from the standards and procedures.	• No documentation standards described
Testing Process (10 points) 7 Lack of knowledge?	• Testing process clearly shows that a quality product will be developed. • Testing process appropriate for project. • Testing plan addresses unit, integration and system levels of testing.	• Testing process somewhat shows that a quality product will be developed. • Testing process somewhat appropriate for project. • Testing plan addresses most of the unit, integration and system levels of testing.	• Testing process does not show that a quality product will be developed. • Testing process not appropriate for project. • Testing plan addresses only addresses one of the unit, integration and system levels of testing.	• No testing process described.
Conclusion	• Retrospective addresses all	• Retrospective addresses	• Retrospective addresses	• No conclusion provided

Criteria	Excellent	Satisfactory	Poor	Unacceptable
(10 points) **10**	three questions • Answers to retrospective questions show deep thought and are from the whole team • Status report provides a detailed view of the team's progress	two of the three questions • Answers to retrospective questions show little thought or are from one person • Status report not very detailed	one of the three questions • Answers to retrospective questions show no thought • No status report	
Overall Document Appearance (20 points) **17**	• Document looks professional • Page numbering starts after the front matter • Introduction appears as page 1 • All tables and figures have captions and are referenced in the text • Appendices referenced in appropriate sections • Document shows consistent use of a template	• Minor inconsistencies in font or formatting • Minor errors in page numbering • One figure or table missing a caption or is not referenced in the text • One appendix not referenced	• Major inconsistencies in font or formatting • No page numbers • Two or three figures or tables missing a caption or not referenced in the text • Two or three apprentices not referenced in text	• No document style used • No captions for figures and tables • Tables and figures not referenced in document • Appendices not referenced
Spelling and Grammar (30 points) **20**	• No spelling errors • No grammatical errors • A range of punctuation used correctly including commas, colons and semicolons	• Minor spelling or grammatical errors • Minor punctuation errors	• Major spelling or grammatical errors • Major punctuation errors	• Spelling and grammatical errors interfere with understanding • Insufficient or lack of punctuation
Clarity and Conciseness of Writing (40 points) **30**	• Words are used correctly and precisely • Vocabulary appropriate for intended reader	• Occasional ambiguity in word choice	• Wordiness *Lots of awk statements*	• Report hard to understand