ECOders

Project: *Colorado Plateau Cooperative Ecosystem Studies Unit (CPCESU) Project Management System*

Requirements Specification

<u>Overview</u>: The purpose of this document is to list out the requirements and our criteria that will determine if we have successfully delivered a successful product that solves out client's problem

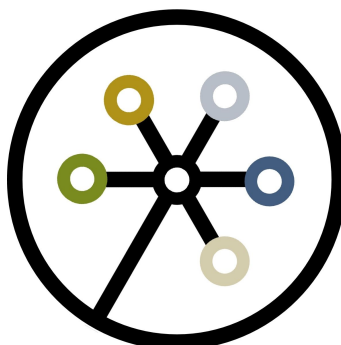|	
Team Members:	***Clients:***
Colton Nunley	Dr. Todd Chaudhry
Joseph Remy, Jr.	Laurie Thom
Jasque Saydyk	
Ana Paula Chaves Steinmacher - *Mentor*	
Northern Arizona University	Cooperative Ecosystem Studies Unit
School of Informatics, Computing, and Cyber Systems	Colorado Plateau

February 1, 2019 - Version 1.3

Approval Signature from ECOders:

_____

Joseph L. Remy, Jr.

Date:

_____

ECOders

Approval Signature from CPCESU:

_____

Dr. Todd Chaudhry

Date:

_____

This page is left blank intentionally.

Colton Nunley   Joseph Remy - *Team Lead*   Jasque Saydyk
crn79@nau.edu   remy@nau.edu   jrs496@nau.edu
v1.3

2

Table of Contents

Colton Nunley   Joseph Remy - *Team Lead*   Jasque Saydyk
crn79@nau.edu   remy@nau.edu   jrs496@nau.edu

This page is left blank intentionally.

1. Introduction

Preserving nature and our past requires a wide variety of work and information to identify the problems, develop solutions, and gain insight into how our world used to be and how it is changing. This is done with a wide variety of methodologies. From investigating archeology sites to conducting zoological surveys, all of these projects help preserve the nature and history of the American Southwest, allowing us to learn more about the impacts we have on our fragile ecosystem.

The Colorado Plateau Cooperative Ecosystem Studies Unit (CPCESU) is a consortium which organizes federal agencies and Native, state, and local governments with non-governmental organizations and universities to complete a variety of these conservation projects. To do this, the CPCESU gets dozens of project proposals, hundreds of modifications, and millions of dollars each year to setup, organize, track, and archive these projects. Since their founding, they have used an ad-hoc approach to project management using email, Microsoft Excel, a Microsoft Access database, and a shared file system drive.

Due to the make-do nature of the forms and data storage, the CPCESU staff are spending up to half of their time at work to track these projects. If the number of projects were to double, the CPCESU staff would find themselves overwhelmed and have no time to spend on their other work for the organization. In addition to this, there is no defined structure for data entry into the database, leading to partially completed rows and a lack of consistent, vital information the organization needs to renew its charter. There is no way for project leaders and organizations to modify their agreements to extend the timeline and allocate more funds without emailing or calling the CPCESU, and the CPCESU staff need to be able to search their dataset and export statistics they need to report to inquiring organizations, NAU, and the CPCESU Director. Lastly, certain parts of CPCESU projects depend on budget spreadsheets, approval documents, final reports and more, which are not directly linked to projects and need to be found manually in a massive file system.

ECOders is a senior Computer Science capstone group formed under the School of Informatics, Computing, and Cyber Systems at Northern Arizona University. Team members include Joseph Remy, Jr. (Team Lead), Colton Nunley, and Jasque Saydyk
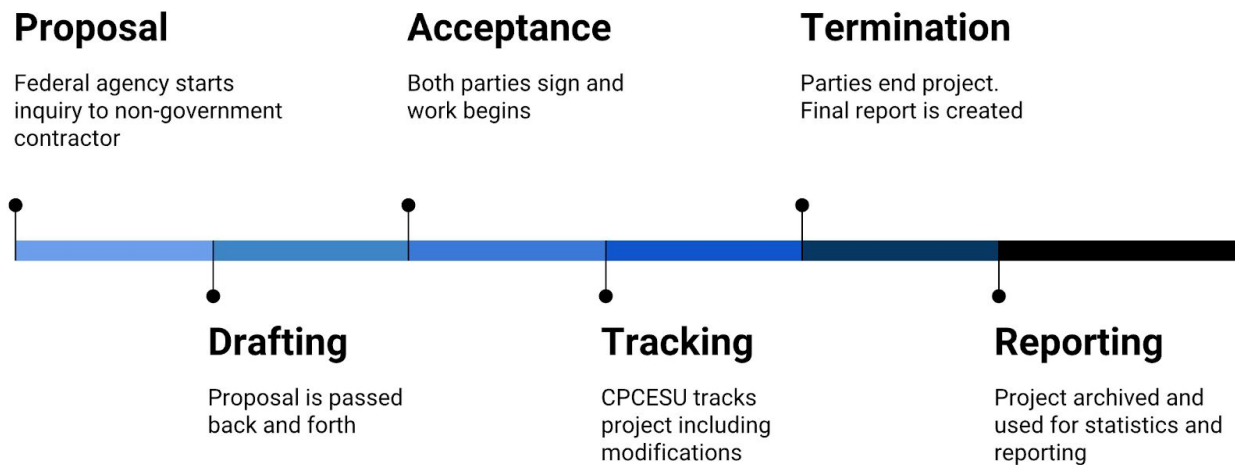
and the Team Mentor is Ana Paula Chaves Steinmacher. We were tasked by Dr. Todd Chaudhry and Laurie Thom with developing a website and database solution for the CPCESU with three essential components: a project management system for the CPCESU; a website for showcasing public projects and information; and a system allowing organizations to develop, submit, and track projects. Our solution, at a glance, shall reduce the time spent tracking and managing the project for CPCESU, allow for organizations to directly submit information into the management system through the use of standardize forms and data import, and must include functionality for quickly and easily exporting data for generating reports.

This software requirements specification begins by detailing out our client's problem and our proposed solution to this problem. After this, we describe every requirement for the project, beginning with the high-level requirements working down to the low-level requirements for the system. Each requirement has a description as to what it entails and some criteria to verify when the requirement has been fulfilled. This is followed by a list of the potential risks with the project, their likelihood, and how we will go about mitigating them. Lastly, we note our project plan for the remainder of the project, detailing any potential risks and mitigation strategies and our expected milestones for the project.

2. Problem Statement

The CPCESU currently uses a combination of various products to create, track, and close projects within their organization. They use email and phone calls to communicate, create, and draft projects; an Excel spreadsheet to track projects through their various phases; a Microsoft Access database to archive the projects and keep track of modifications to projects, though it is currently not fully populated or functional for tracking changes; and a shared file system to store the files of the CPCESU as they draft and archive the final reports of the projects.

Figure 2.1: CPCESU Project Workflow

Proposal

Federal agency starts inquiry to non-government contractor

Acceptance

Both parties sign and work begins

Termination

Parties end project. Final report is created

Drafting

Proposal is passed back and forth

Tracking

CPCESU tracks project including modifications

Reporting

Project archived and used for statistics and reporting

As for the CPCESUs workflow (see Figure 2.1 above), projects are initiated by the federal agency soliciting the non-federal partner. The project is then assigned a Principal Investigator, who negotiates with the organization and the CPCESU as they draft the Task Agreement for the project, which is a contract for how the project is conducted, funded, and reported. When the partner and the federal agency both sign the contract, the project is then accepted and enters the tracking phase. During the tracking phase, projects are modified by the federal partner, which could be anything with the project, but is mainly funding, personel, or the period of performance. When a final report for a project has been completed or a project has been deallocated resources, then the project is closed, entering the termination phase as it is archived into the CPCESU database. While archived, the CPCESU may need to access the project's data again to report on it, give it to a requesting agency, or to distribute some select information to the public.
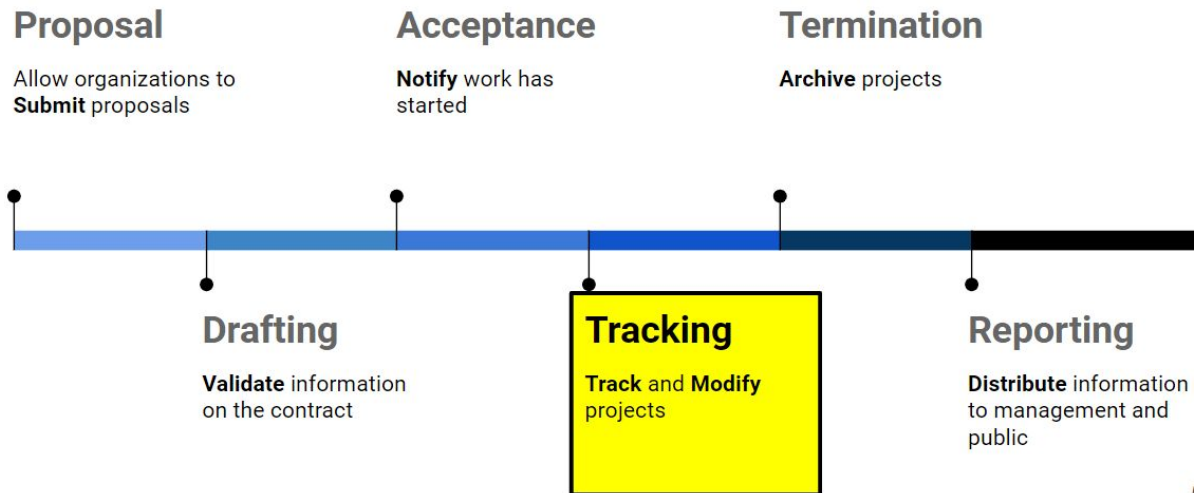
Proposal

Allow organizations to
Submit proposals

Acceptance

Notify work has
started

Termination

Archive projects

Drafting

Validate information
on the contract

Tracking

Track and **Modify**
projects

Reporting

Distribute information
to management and
public

Figure 2.2: CPCESU Project Workflow with highlighted, problematic stage

The main issue with the CPCESUs workflow is with the tracking phase, as shown in Figure 2.2, as they receive hundreds of modifications for projects each year that modifying the budget, time, or anything else regarding the projects. CPCESU suffers from being unable to track these effectively. They already spend up to half their time at work tracking these projects, but with their work being scattered across several programs, there is no one system to make these modifications. If there has been a mistake, they cannot easily look at the history of a modification to correct, which has steadily lead to the problem of a database that is never up to date with the current projects, which fixing would only consume more and more time to do.

This has lead to the current situation of having to recharter the organization with NAU using inaccurate budget numbers from this outdated database. Tens of millions of dollars go unreported. This puts their organization in jeopardy, as NAU wants them to see at least a 50% indirect rate or overhead of what the project brings in, rather then the 17.5% they currently report. This requires CPCESU to sell the 17.5% as a large throughput of projects, high value for the 17.5%, and providing work to students.

The CPCESU also has some other minor issues that need to be handled by any solution that is delivered. They need a mechanism to allow organizations to submit projects including automated and manual data entry, as organizations will sometimes begin projects without the CPCESU's knowledge. This leads to situations where the CPCESU is made aware of the project when it is completed, which may be due to the time cost of communicating with the CPCESU, which we seek to reduce.

They also need to validate the information coming into their organization to ensure the database stays correct and doesn't contain common mistakes. As their database has

been survived through the decades, it has slowly been filled with more and more mistakes and design flaws that need to be corrected. Small issues like not upper casing names and mistyping in an organization's name then makes it impossible to search and find information in the database.

Lastly, they need a way to distribute general information to the public and organizations. They currently distribute information about their projects with links to PDFs on their NAU website, which is not searchable or maintainable in the long run. In addition to this, various organizations call the CPCESU requesting a list of projects they have done with them, or information regarding previous projects are pulled up in a meeting. They currently fulfil these requests by running premade queries on their database and exporting the results to Excel.

3. Solution Vision - *Summit*

Our solution to the CPCESUs problem, nicknamed "Summit", will be a web application with three primary sections: a public facing front end that distributes projects and informs the public about the CPCESU, a section for organizations to submit project proposals and modifications to projects, and a backend for the CPCESU to manage, track, and modify their projects. All of this is archived into a robust, easy to access, and maintainable database, as shown in Figure 3.1 below.

Figure 3.1: Solution Diagram
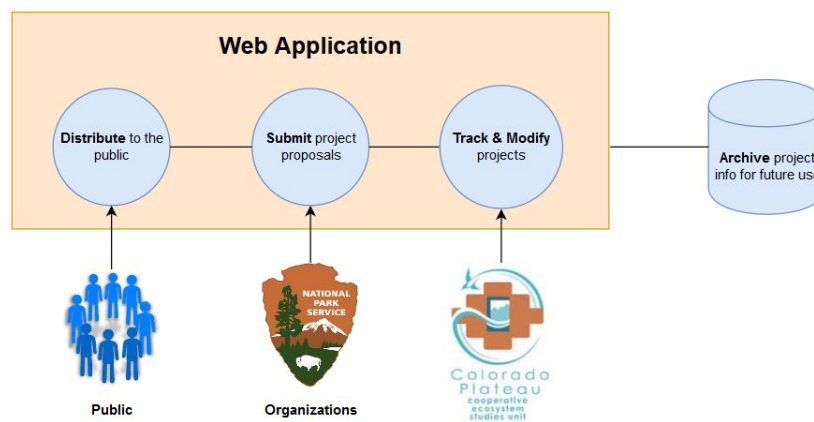
The overall functionality of our solution will have a very heavy focus in supporting our client's workflow. We want them to be able to hop on our web portal, query a database for project information, and get quick and easy to read results. This also involves creating a seamless transition between project details without losing information. Finally, storing data from previous projects for analysis and statistics that will be used to support future projects is also an important aspect to our vision. The production version of the website will be hosted using NAU ITS, which will be in a Red Hat Enterprise Linux 7 virtual machine. The website itself will be built using Django v1.11, which is the latest long term supported version of the framework till 2020. The database back-end will be PostgreSQL 7 and Gunicorn will be used as the WSGI HTTP server for the website.

Our solution will fix the client's problem by providing each stakeholder to this website a unique way of accessing the website that addresses their needs. Most importantly, it will decrease the workload with regards to finding and analyzing project data.

4. Project Requirements

To deliver a product that matches the solution vision, we have developed various requirements. There are various different types of requirements:

- User Requirements
- Functional
- Non-functional
    - Environmental
    - Performance
    - Constraints

Each of these requirements vary in nature, with each one feeding into the next type. User requirements are the top-level requirements.

User Requirements (UR)

We created the following user requirements to define our business requirements that will be developed into technical requirements. They are written as user stories to help quickly and succinctly convey the broader requirements of this solution.

UR.01 - As a CPCESU Staffer, I want to track projects so that I understand, at a glance, what the project status is.

UR.02 - As a CPCESU Staffer, I want to be notified to review and approve of project modifications.

UR.03 - As a CPCESU Staffer, I want to ensure correct data so that common errors are not entered into the database.

UR.04 - As a CPCESU Staffer, I want to autofill from a document so that I don't have to type in all of the data.

UR.05 - As a CPCESU Staffer, I want to generate and export data from querying the database and a project search page.

UR.06 - As a CPCESU Staffer, I want to create and import projects on behalf of my partners and federal agencies.

UR.07 - As a student, I want to be able to search through CPCESU projects so that I can find research I am interested in.

UR.08 - As a person with disabilities, I want to be able to use the CPCESU website so that I can get involved with the CPCESU organization.

From these user requirements, we have created the following functional and non-functional requirements, which are of way of conveying how our solution should be built and simplifies tracking of progress. Certain requirements have a higher level of abstraction and can include lower-level requirements. Overall, all of our requirements fall into five hemispheres of requirements or domains, namely:

1. Creating a core, functional project management system
2. Developing user permissions and functionality for organizations and partners
3. Expanding the application with public-facing pages
4. Building an intuitive, universally accessible website
5. Adding additional capabilities for data entry and project tracking

By organizing our functional requirements into domains, it makes it easier to classify and prioritize our work and how we should create a timeline for our project. Each domain contains functional high-level requirements and nested lower-level requirements, constraints, and so on. In the following sections, we will be addressing each requirement, its type, and how it is important to our project and to our client.

Functional Requirements

In this section, we will cover our core functional requirements and have various layers of granularity to further specify the technical aspects of the solution that shall be implemented.

Domain 1: Creating a core, functional project management system

Domain 1 is about creating the core of the project and the overall minimum viable product: a project management system that works for only the CPCESU staff. Specifically, these are **Software Requirements Specifications (SRS): UR.01** through **UR.05**.

F1.00 - Projects (also named Task Agreements) shall be tracked through their six phases: Proposal, Drafting, Acceptance, Tracking, Termination, Reporting

> **F1.01** - All of the projects shall be listed out to be viewed at a glance

> **F1.02** - Projects shall be marked so that they can be identified through five of their phases, Proposal, Drafting, Acceptance, Tracking, and Termination

> **F1.03** - Projects shall have fields so that the following fields can be identified:

- Status - Executed, Submitted, Pending, Uncertain
- Federal Agency
- Alternate Research Coordinator / CESU Representative
- Parks / Locations
- Partner
- Project Title
- Federal Point of Contact
- Partner Principle Investigator
- Initial Funding Amount (USD)
- Current/Final Funding Amount (USD)
    - Calculated using Initial + modification amounts
- Tentative Start Date
- Tentative End Date
- Student Support
    - Graduate
    - Undergraduate
    - Both Groups
    - None
    - Unknown
- Youth / Veteran Involvement (Dropdown)
    - Youth / Young Adults

- ○ Veterans
- ○ Both Groups
- ○ Not Applicable
- Sensitive Data (Yes or No)
- Discipline
- Type
- Research and Development Fields
  - ○ Type of R&D (required)
- Date Project Initially Received
- Date Review Comments Sent
- Date Task Agreement Approved
- Data Executed
- Date Actual Start
- Date Actual End
- P Number
- Fiscal Year
- Project Abstract
- Keywords
- Final Report (Yes or No)
- Permission to Share Final Report (Yes or No)
  - ○ Sensitive field removes this field

F1.04 - Projects shall include the optional fields:

- Agreements Tech Representative if NPS selected
  - ○ Project lead - defaults to Point of Contact if empty
- If Type of R&D is applicable, include the following:
  - ○ Source of Funding
  - ○ Monitoring (Yes or No)
  - ○ Scientific Method (Yes or No)
  - ○ Fields and subfields
  - ○ Requires IACUC Review / Concurrence (Yes or No)

F1.05 - In a dedicated project search page, projects shall be filtered and sorted using the following fields:

- Project Status
- Project Title

- Federal Agency
- Partner Organization
- Type
- Discipline
- Fiscal Year
- Budget

F1.06 - Projects should allow users to upload attachments to justify work

    F1.06.01 - This may include the following document types

- Supporting Documents - .pdf, .doc, .docx, .jpg, .png, .gif, .tif
- Budgets and Personnel Allotment - .csv, .xls, .xlsx

F1.07 - At a minimum, the project dashboard for CPCESU shall include the following fields for each active project:

- Project Status
  - Location on global workflow
- Point of Contact
- Principal Investigator
- Budget (including any modifications)
- Indication of Modification Status
  - View will say whether or not a project has modifications pending or requesting review

F1.08 - Adding and changing project fields or status shall be logged to show who last added and updated projects, the fields changed, and when the change occurred.

F2.00 - Modifications to a project shall be tracked 💬

F2.01 - When a modification is submitted by the organization outside of the system, a CP Staffer shall be able to submit the modification to *Summit*

    F2.01.01 - There shall be a button on the project details page that can be clicked to add modifications

    F2.01.02 - This button will redirect to a custom view where the modification can be filled in by the appointed CP Staffer(s)

F2.02 - The CP Staff shall be able to edit and archive modifications

F2.02.01 - When adding and editing modifications, these changes shall be logged.

See **SRS: F2.03** for logging procedures

F2.02.02 - When adding or editing modifications, the following fields will be the same as the original project fields in **F1.03** and **F1.04**. This is will also include a status for the modification (Drafting, Approved, Executed).

F2.03 - The result of this modification, adding or editing, shall be logged

F2.03.01 The system shall automatically keep track of all modifications to show revisions

F3.00 Projects shall be terminated based on the following requirements.

F3.01 - Project is terminated if the project has reached its final end date and/or if a final report has been submitted (if project completes early).

F3.02 - The project termination process will have its own page to allow the CP staff member to upload a final report and get a review of the project before closing it.

F3.03 - Project termination shall be the same general process including:

- Email and in-system notifications shall be sent to the CP Staffer assigned to project
- The project will be in the Archived state and the email will let the CP Staffer know if there is a report or not

F4.00 Data coming into and out of the system shall be validated for data quality and assurance

F4.01 - All user input fields shall have client and server-side verification

F4.02 - The following fields shall have dropdown, search, or auto-fill capabilities to limit user input errors:

- Partners
  - Principal Investigator
- Type

F4.03 - The following fields shall have the appropriate user field and verifications based on its type:

- Expected Start Date - Date
- Expected End Date - Date
- Budget - Number
- User constraints on all forms
  - Not all text boxes, but the best constraints are using dropdowns and autofill

F5.00 Data from projects shall have the ability to be download and/or exported

    F5.01 - CP Staffers may print or save the project details page as a PDF

    F5.02 - CP Staffers shall be able to download project attachments

Domain 2: Developing user permissions and functionality for organizations and partners

Domain 2 involves extending the functionality of the project management system to allow for federal agencies and other partners to access the system. This includes segmenting the user base into groups and limit access based on a lowest-level permission system and hierarchy. These functional requirements are based on **SRS: UR.06** and **UR.07**.

F6.00 - The web application shall be made up of user groups to provide access to the system, customized functionality, and the separation of permissions

    F6.01 - The solution shall include, at a minimum, the following user groups:

        F6.01.01 - CPCESU - Top-level management group

- Administrators - Examples: Todd and Laurie
- Staff - Example: Rebecca
- Read-only access

        F6.01.02 - Federal Agencies

- Admins - Example: primary Point of Contacts
- Staff
- Read-only access

        F6.01.03 - Non-governmental Partners

- Admins - Example: primary Point of Contacts
- Staff
- Read-only access

F6.01.04 - Public - Anonymous access

- No users or groups, exclusively read-only access

F6.02 - The CPCESU administrators shall be allowed to add, modify, or remove the groups and users of the system

F6.02.01 - The CPCESU administrators shall follow the following requirements for modifying user groups:

- CP administrators shall be able to add, modify, and remove users and groups through the admin dashboard built-into the web framework
- Partners and federal agencies may have an application process built into the web application to apply

F6.03 - Federal agencies and partners shall be allowed to add, modify, and remove users from their own group

F6.03.01 - Group administrators shall be allowed to invite new users, modify users and group details, and change user permissions

F6.04 - The public (anonymous users) shall have no access to the system except for public facing pages as defined by **SRS: F8**

F7.00 Each type of user may have permissions and privileges based on their user group and status. At a baseline, the foundation shall be set based on the following:

F7.01 - Users shall be allowed to view/query/search database

F7.01.01 - CPCESU and their federal agencies, partners, and other CESU users shall be allowed to view/query/search including:

- The database for specified projects
- Fiscal year, federal agency, and/or partner
- Past, present, proposed, and cancelled projects
- Any sensitive information relating to each project

F7.01.02 - Public users shall be allowed to view/query/search by:

- General, non-sensitive project information as outlined in
SRS: F8

F7.02 - Only the CPCESU user group shall be allowed to export data from all
projects and user groups

- Export via PostgreSQL web command tool
- Export as CSV / Excel file via the web application

F7.03 - Only the CPCESU administrators and federal agencies administrators and
staff may start a project proposal

> **F7.03.01** - CPCESU shall enforce a standard form for all federal agencies
> using the required and optional fields from **SRS: F1.03** and **SRS: F1.04**
>
> **F7.03.02** - Federal agency administrators and staff may have access to
> this form to start a new task agreement
>
> **F7.03.03** - The designated partner, its administrators, and appointed staff
> members (usually Principal Investigator(s)) may have access to this
> project proposal once assigned by a CPCESU administrator or designated
> staff member

Domain 3: Expand the application with public-facing pages

For Domain 3, we extend the user groups to the anonymous access or the public
front-end. This will allow users to access publicly accessible information without having
to register or log in. This also creates an embeddable project search page. The goal of
this domain is to satisfy **SRS: UR.08**.

F8.00 The public shall be allowed to view the following information as explained below:

> **F8.01** Homepage shall contain information pertaining to the CPCESU loosely
> based on their current website at https://in.nau.edu/cpcesu, but at a surface
> level. This means brief information as defined in **F8.01.01**, no documents or
> attachments, or redundancy with the main CPCESU website.
>
> > **F8.01.01** - Homepage shall contain the following:
> >
> > - Summary of CPCESU

- Contact information
- Link to current NAU website as noted in **SRS: F8.01**
- Link to project search

F8.01.02 - Homepage may contain the following:

- Highlighted projects
    - May be based on the most recent start time
    - May be CPCESU administrator or staff selected

F8.02 - The project search page shall allow the public to search through the CPCESU project management system without registering or logging in

F8.02.01 - Each project on the search page shall contain the following information as outlined in Figure 4.1:

Figure 4.1: An example of what the public can view

Task Agreement	P17AC00181
Project Title	African American Homesteaders Historic Resource Study for HOME and NICO (HRS)
Partner Institution	University of Nebraska-Lincoln
Federal Agency	National Park Service
Project Start Year	2017
Project Status	Active

F8.02.02 - The search criteria for this page shall contain the following fields as outlined in Figure 4.2 below. We will also allow the search box (left side of the figure) to also query through the project summary / abstract.

Figure 4.2: An example of what the public can search and filter on

> **F8.02.03** The current CPCESU Wordpress site shall be allowed to embed this search page on its site

Domain 4: Building an intuitive, universally accessible website

Domain 4 involves the general appearance and responsiveness of the website. Primarily, this encompasses the requirements for software accessibility. This domain exclusively covers **SRS: UR.09**.

F9.00 - The website and any of its software shall implement software accessibility for all users, regardless of disability, impairment, etc.

> **F9.01** The solution shall adhere to the following accessibility standards:
> - NAU's Web Accessibility Standards
> - W3C's Web Accessibility Standards

Domain 5: Expand on additional capabilities for data entry and project tracking

Lastly, Domain 5 involves requirements that may be optional to the project, but can ultimately extend the solution's functionality and use. This involves things such as data

importing, customizable workflows set by federal agencies, and so on. This last domain expands on all of our user requirements, **SRS: UR.01** through **UR.09**.

F10.00 Our web application shall make it possible to import data directly from user uploaded documents as specified below:

> **F10.01** - This system may allow for both new and older projects to be uploaded

> **F10.02** - This system may allow for scanning and optical character recognition for the following use cases:
>
> - Creating a new project
> - Archiving an old project that is already terminated
> - Submitting a modification or request to terminate an existing project

> **F10.03** - Minimal viable product shall use the Grant and Cooperative Agreement form to automatically fill in the project and modification forms.

F11.00 - The solution may encompass a more detailed project details page for the project search capability explained in **SRS: F8.02**

> **F11.01** - The projects on the search page shall have a way to direct the user to that specific project's public details page

> **F11.02** - The public project details page shall include, at a minimum, the following details as shown in Figure 4.3, unless the field is marked as sensitive:

Figure 4.3: An example of what the public can view on the project details page after searching

Project Title: African American Homesteaders Historic Resource Study for HOME and NICO (HRS)	**Location(s) of Project Activity:** Homestead National Monument and Nicodemus National Historical Site
Task Agreement: P17AC00181	**Partner Institution** University of Nebraska-Lincoln
Project Status: Active	**Federal Agency:** National Park Service
Project Start Year: 2017	**Student Involvement:** Unknown
Project Funds $175,224.00	
Principal Investigator(s) Richard Edwards	**Federal Agency Contact** Robert Marcell

Project Abstract
This project will explore the history, circumstances, and outcomes of African American homesteading experiences in the Great Plains. It provides new scholarship on this little known history by focusing on three historic venues of African American homesteading: (i) DeWitty, Neb.; (ii) Nicodemus, Kan.; (iii) Empire, WY.; and it will also include, if possible, a fourth focus, (iv) isolated African American homesteaders (such as the Shores family in Custer County, Neb.) elsewhere in the Great Plains, if a sufficient number of such homesteaders can be identified. The purpose of the research will be to identify who the homesteaders and their families were, what land claims they made, how successful they were in making claims and gaining ownership, and other information about their educational, cultural, social, recreational, and other life experiences as homesteaders. An effort will be made to document the specific landscapes occupied by African American homesteaders. This project will also benefit the interpretive programs at both Homestead National Monument of America (HOME) and Nicodemus National Historic Site (NICO) in addition to filling a gap in the scholarship of this important national homesteading story.

Project Modification 1
Year: 2017 Task Modification Number: 0001
Funds: $0.00

Project Modification 2
Year: 2018 Task Modification Number: 0002
Funds: $66,770.00

F12.00 The solution may allow for each federal agency to control its own fields when creating a new project, adding a modification, or having a partner close the project

> **F12.01** - The default forms used by all parties shall be the ones set by the CP administrators to assure that required CPCESU information is obtained

F12.01.01 - All custom forms set by the federal agency's administrators shall include the required CPCESU fields and use the CPCESU forms as templates

F12.01.02 - Any conflicts between the forms shall use the CPCESU version of the field

F12.01.03 - Federal agency forms shall not remove any of the required fields set by the CPCESU

F12.02 - Additional fields that are set by the federal agency must be from the global type list included below:

- Text
  - Single and multi-line
- Number
  - Can be floats (like budgets with dollars and cents) or integers
- Datetime
- User selector
  - Based on autofill or search capabilities for the federal agency's users

F13.00 The solution may allow the public users to grab public, non-sensitive data from the CPCESU without directly interacting with the database

F13.01 - This functionality shall be read-only and rate limited by the web framework to reduce traffic where possible

F13.02 - This functionality may include private access for other CESUs to access and read data

F14.00 Our web application may have functionality that will support the search for experts in all of the non-federal partner groups

F14.01 - Users may have the ability to find experts based on fields of study which may include

- Anthropology to Zoology
- This functionality is not accomplished before by any other CESU

Non-functional Requirements

This section will cover the various non-functional requirements that will affect the project overall, but do not specifically detail features. These include environmental requirements, constraints, performance requirements, and so on.

NF1.00 Data shall always:

- Immediately update in database
- Be done transferred live and between the CPCESU web server and the institutional database server to make the changes
- Take less than five seconds to get or post, excluding file uploads and dependent on current CPCESU server or institutional network and database usage

NF2.00 Our web application shall have security that protects sensitive information

NF2.01 - Sensitive information includes, but is not limited to:

- Budgets
- Content of final reports
    - Local information such as archaeological sites, endangered species information, etc.

NF2.02 - These guidelines and plans shall be outlined in the Security Plan described in **SRS: NF9**

NF3.00 The CESU and federal agencies may have constraints or specifications that our web application including, but are not limited to:

NF3.01 - The solution shall not be run by the end users or require users to download an executable of any kind. Federal computers used by agencies do not allow external software

NF3.02 - The software shall allow for the eventual national functionality to allow for the sharing of data amongst the CESU network and the public

- The data may include:
    - Public project data
    - List of federal agencies and CPCESU partners
- This data shall be read-only

○ The functionality for this is described in **SRS: F13**

NF4.00 The deployment shall be executed with the following constraints and specifications as per NAU ITS and CPCESU:

NF4.01 - NAU ITS shall host the production server as per the agreements between NAU, the host organization, and the CPCESU

○ The project shall use NAU ITS supported tech only
○ ECOders shall adhere to providing any documentation requests as per NAU ITS's policies, code of conduct, etc. These are outlined below as **SRS: NF8** and **SRS: NF9**

NF4.02 - NAU ITS shall provide a domain name based on CPCESU's request

○ This must be locked in for SSL certificate for secure communications between the end users and our solution.
○ This will be taken care of (and paid for) by NAU ITS

NF5.00 Authentication and sensitive data should be contained with:

● The need to use blocking defaults and "best practices"
● Sensitive data access should have least permissions
● This will be outlined in the Security Plan in **SRS: NF9**

NF6.00 Growth of our web application should allow for:

● For up to and including approximately 100 users for now
● The ability to add additional resources if need be
● The growth strategy for this application shall be outlined in the Maintenance Plan as per **SRS: NF8**

NF7.00 Institutional database constraints set by NAU ITS shall allow for:

● Database being located on a seperate host
● All backing up and maintenance done by NAU ITS
● BLOBs as the file storage

NF8.00 A Maintenance Plan shall be in place before the solution is delivered to the client

● This plan is required to ensure continuous functionality of the web application
● The plan will keep in mind that ITS does nightly security updates as well as major seasonal updates

● Required by NAU ITS to make sure the machine and Summit are maintained

NF9.00 A Security Plan shall be created and be in place before the solution is delivered to the client

● This is required by NAU ITS to ensure protection of sensitive data in the database, software security, and so on.

NF10.00 Our web application shall implement Web Server Gateway Interface (WSGI) to serve multiple users at once

NF10.01 - Green Unicorn (Gunicorn) or Apache may be the WSGI technology that our web application uses. Others are allowed depending on NAU ITS

NF10.02 - Our WSGI implementation must be NAU ITS standards and have an included segment in the Maintenance Plan referenced in **SRS: NF8.00**

NF11.00 The user interface shall be responsive and use the principles of material design

NF11.01 The front-end shall respond within 250 ms to any user input on client side

NF11.02 The front-end shall follow the material design principles set out by Google at https://material.io. Specifically, this includes:

● Creating a unified user experience
● Customizability tailored to the users of the product
● Implement good design principles based on Don Norman's Principles of Interaction Design:
    ○ Visibility - the overall appearance of the user interface
    ○ Affordance - what the user can do with the interface, both the expected and unexpected
    ○ Mapping - what labels are needed or present

5. Process for Requirements Modification

Though we have define the requirements to the best of our ability for this project, it is nearly impossible to define all of the requirements up front. As development begins and stakeholders get a clearer idea of what is possible and what they need, change requests will naturally arise. In addition to this, regulations change, opportunities arise, and business needs change over the course of development, which also demands that we be flexible and be willing and able to change our course during development. However, change always has a price and it maybe necessary to modify existing requirements to accommodate the changes.
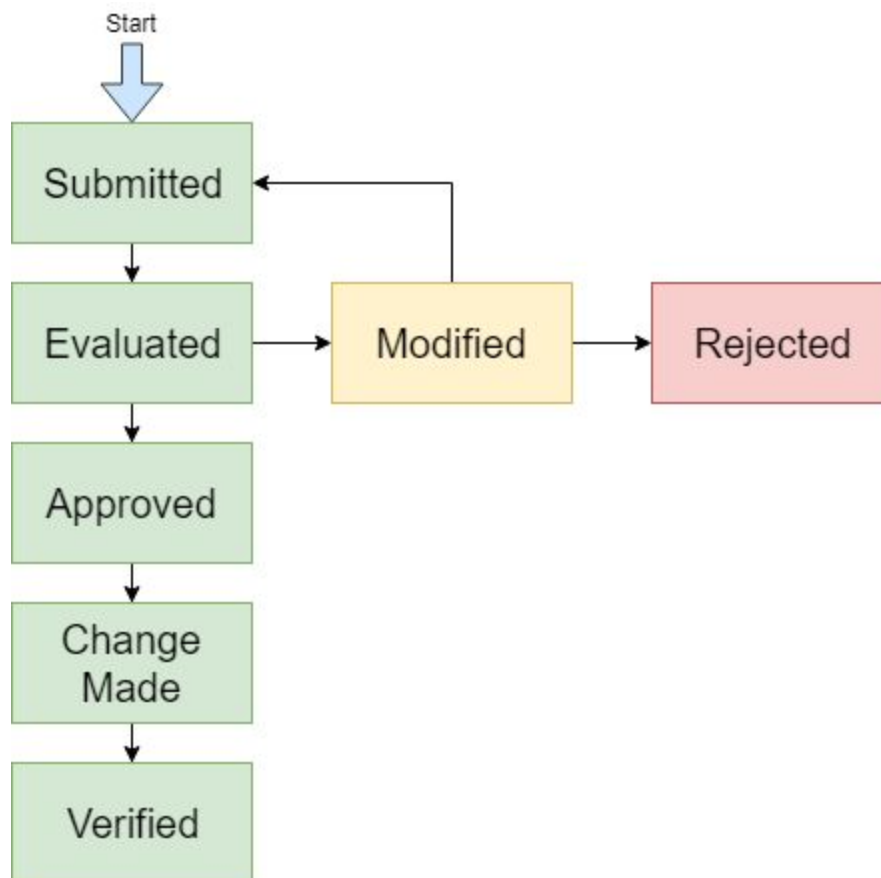

Figure 5.1: Flow for modifying this document

Any change that is requested shall go through the following process:
(Also shown in Figure 5.1)

1. Any change will need to be submitted, either through email or any other recorded communication method
2. The change will then be evaluated for the time required to implement the change
   a. If there is some issue with the proposed change, the team will attempt to modify the request to be able to fit it into the project
3. The change is then either approved or rejected
   a. For small changes to low level requirements, team consensus is all that is need to approve the changes
   b. For large changes that would involve multiple sprints to complete, team mentor approval is also required along with team consensus.
4. If approved, the change is reflected in this document and resigned by both parties
5. After resigning and approving the updated requirements document, it is then implemented and verified by the client

With this simple process, we hope to accommodate all potential changes to Summit that may be proposed throughout the development of this product.

.

6. Potential Risks

During our requirements gathering and technological feasibility studies, we have determined a few key risks to this project's success and overall impact. These were documented as we went through both processes and conducted general research including reading about software engineering (namely the risks and mitigation strategies). It is important to outline the risks and ways to overcome them, because they can ultimately lead to the project not functioning or completely failing.

All of our risks are high-level and can encompass many different, low-level issues. We have also assigned a percentage of occurance *at the worst case* (probability of happening without mitigation) and a general low, medium, or high impact rating.

Risk 1: Learning Curve Too High | 25% probability | *High*

The risk here is that, as we work on the project, we may develop a solution that is not specifically tailored to the CPCESU or that people in the organization may find it difficult to use the software when just dropped into it. This may limit the adoption of our product into the organization, or in a worst case scenario, may lead to our solution being entirely rejected. This also has the highest probability of happening, as this problem will develop rapidly over time, and by the time we realize that it is a problem, it maybe too late in the development lifecycle to change the product.

We plan on overcoming this with two simple strategies: early prototyping and constant deliverables for our client to get feedback on the requirements we are implementing and, overall, to assure that we are delivering the solution they need. To generate these deliverables, we will be doing 2-week release periods for production and rapid release to our development server using Continuous Integration and Continuous Delivery methodologies.

Risk 2: Overlooked or Missed Requirements | 10% probability | *Mid to High*

While we are working with our clients, there is always an amount of risk that we may have missed a crucial requirement for the system. As we develop the requirements, we will hopefully diminish this risk, but there is always a possibility that the client mentions something that may have been a big requirement that we did not catch. As the project

gets developed, the client will bring these additional requirements to the forefront, which we want to remain ready to accommodate if possible.

To minimize this risk, we plan scheduling a buffer period into our schedule for this project to allow work on any additional requirements. In addition to this, we have included procedures in this document to allow for modifications and new requirements, although this may require changing or limiting certain other requirements that already exist in the document. As always, we stress that we want to give our client a product that will satisfy all of their needs in the given time we have, so anyway we can go about doing that, we will.

Risk 3: Security Breach | 5% probability | *Mid to High*

Our client works with some atypical, sensitive information concerning the locations of endangered species and archeological sites. This information is controlled under the Archaeological Resources Protection Act of 1979 (ARPA), which states that information on the nature and location of resources on public and Indian lands must remain confidential if its release may harm the resources. Violating the ARPA the first time entails a fine of $300,000 and up to one year in prison. To better protect the history of America and protect our client from violating federal law, it is imperative we build a solution that is secure against internal and external actors. While the risk of this happening is relatively low, as the CPCESU is not a large, well-known organization, the impact of a security breach can, in the worst case, cost hundreds of thousands of dollars and irreparable harm the reputation of the CPCESU.

Our solution is to ingrain modern security practices into our development, to ensure that we are delivering a secure product to our client. On top of this, we plan on performing penetration testing on a clone of our website after every two sprints to determine the security flaws with our solution. The results of this penetration testing will be a report that states different ways to access the website, and the severity of the breach. Our goal is to deliver a solution that is secure against all high impact security breaches and minimizes and is able to track low and medium impact breaches as much as possible, though this may be compromised for usability in noted cases.

Our client will not prevent uploading sensitive documents to the system and keep a local backup only. Our system, when the project is marked as sensitive, will not allow final reports, attachments, or agreements to be uploaded unless marked as "Not sensitive" and allow for public users to request additional information via the project management and notification systems.

7. Project Plan

The project plan is an essential part to every software project as its primary purpose is to notice when the project is veering off-course as early on as possible, which allows for ample time to readjust the project and the final product.

To develop the CPCESU Project Management System named Summit, we plan to work in software sprints of two weeks, implementing at least a core feature every two weeks for the website. Throughout the sprint, the development server will host the website in its most current form and, at the end of the software sprint, a production release of the software is made to the production server. This will allow the client ample time to review the website and request any changes and report any bugs during development. We will have a total of seven milestones throughout the semester up until spring break.

When Spring break is over (March 24th, 2019), we should have all major features of the website completed and the product will enter into a production environment as much as possible for the client to help determine the existing flaws with the product. These last three sprints will be called the Acceptance Testing sprints, and their purpose is the get the product fully ready to be adopted by the client. At the end of the final Acceptance Testing sprint, the product is delivered to the client and they will determine if they will accept the solution or not.

This is all visually outlined in Figure 7.1 on the next page.

Figure 7.1: Gantt chart of our timeline for Summit development as of November 18th, 2018
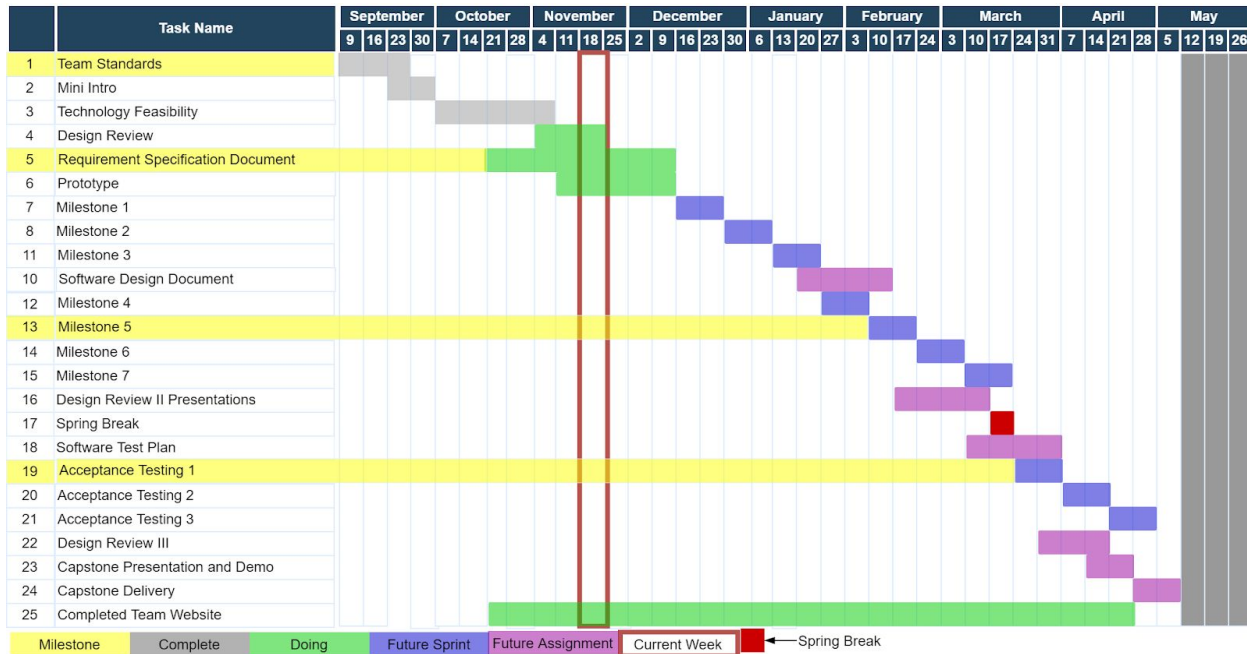
A break down of the goals of each milestone are given below. In general, each major feature of the website has a setup, complete, and verified phase. The setup phase consists of a beta or partial implementation of the feature, complete phase consists of the feature completed, and verified phase consists of getting the feature looked at by the client and adjusting it to their comments if needed.

Sprint Name	Goal	Requirement IDs
Milestone 1	- CI/CD setup - Overall sitemap setup - Login/accounts setup - Forms setup	**F6.00**
Milestone 2	- Implement core project management ideas - Notifications barebones complete	**F1.00** **F13.00**
Milestone 3	- Additional Project management features and testing - Modification system setup - Forms completed	**F4.00** **F12.00**

Milestone 4	-	Autofill forms setup	**F2.00**
	-	Project management verified	**F3.00**
	-	Modification system complete	**F10.00**
	-	Data export setup	
	-	Search setup	

Milestone 5	-	Data export complete	**F8.00**
	-	Search complete	**F11.00**
	-	Modification system verified	
	-	Autofill forms complete	
	-	Public facing website complete	

Milestone 6	-	Data export verified	**F5.00**
	-	Search verified	**F9.00**
	-	Autofill forms verified	

| Milestone 7 | | *Buffer - Used for extra requested features and lagged schedules* | |

8. Conclusion

Our environment is an important part of who we are as human beings. As Carl Sagan once said, "it underscores our responsibility to deal more kindly with one another, and to preserve and cherish the pale blue dot, the only home we've ever known." It is our innate responsibility to protect and preserve the Earth. Solving CPCESU's problem with their current workflow and project management system would give them the ability to focus their time and effort on more pertinent matters instead of trying to find lost data. By building Summit for the CPCESU, we can support their passion for the preservation and protection of the Southwestern United States.

To review the problem, the CPCESU currently does not have a solid project management system that will allow them to store and retrieve data properly. Our clients would benefit greatly from an improved workflow and environment that is easy to use and maintain. The solution to this problem will be our project management system, Summit. Our web-based solution will provide our clients with a simple to use interface and streamlined database access, ultimately giving them the ability to effectively manage past, current, and potentially future projects. By providing data security, easy database queries, as well as a user management system with privileges, our clients can focus more of their time on important projects with quality trusted support from our project management system.

By creating this document, we have laid out the foundations of what is to come over the next few months. We will treat this document as our guide as we move into the development phase, that way our clients can know what we are working on as well as gauge our progress. By defining these requirements in a formal document, we have shined light on our path to creating a product that our client not only needs but is excited about.

As ECOders, we have come up with a plan that will allot enough time for our team to bring a good and working product to our clients in May. Throughout this process of building a plan, we came across a few key insights. One of which was that our biggest hurdle was going to be deployment of our product. We established a positive communication with NAU ITS, who will be the host of our product, and agreed on the technology that will support our system. Another key insight we established early on

was that the client wants the product to be useful for the next five years. This means that as we develop our product, we must be providing future maintainers with enough valuable information to understand the technology and functionality of our code. With all the careful thought and determination that we have been fostering over the past few months we have complete faith that we will give the CPCESU a project management system that they deserve.

9. Appendices

In the below sections are the glossary for looking up word and phrases and their definition in terms of this document and the domain of both ECOders and CPCESU. Following that are any auxiliary documents or sections that are referenced and/or supported in this document such as reports further detailing requirements, diagrams, processes, and so on.

Glossary

API - Application Program Interface is a set of routines, protocols, and tools for building software applications

BLOBs - Binary Large OBjects, a collection of binary data stored as a single entity in a database management system

CPCESU - Colorado Plateau Cooperative Ecosystem Studies Unit, which works with a federal network to help federal agencies contract ecological projects to non-governmental partners.

CI/CD - Continuous Integration/Continuous Development is a development style that will allow for more frequent and reliable code changes

ECOders - An NAU 2018-19 computer science capstone team

HTTP - Hypertext Transfer Protocol - Underlying protocol for sending and receiving data over the World Wide Web or Internet

NAU ITS - Northern Arizona University Information Technology Services

OCR - Optical Character Recognition - the conversion of typed, handwritten, or printed text into digital text

PostgreSQL - an object-relational database management system with an emphasis on extensibility and standards compliance

REST - REpresentational State Transfer is a software architectural style that defines a set of constraints to be used for creating web services

SSL - Secure Socket Link

SQL - Structured Query Language, used to communicate with a database

WSGI - Web Service Gateway Interface, allows for multiple users to asynchronously use a web application at once

WordPress - a free and open-source content management system based on PHP and MySQL

Attachments

There are no supporting documents or attachments presently.

Colton Nunley   Joseph Remy - *Team Lead*   Jasque Saydyk
crn79@nau.edu   remy@nau.edu   jrs496@nau.edu