

Project: P3 - Interactive Lifecycle Documentation Development System

Lead: Jeremiah Butler

Members:

Peter Magalhaes

Ari Ushani

Kevin Palmer

Project Overview

- Problem Description (in your own words)

NUWC is using Microsoft Word to create and modify lifecycle documents, along with a spreadsheet to manually track each requirement throughout the design, development, and testing phases. The problem is that many existing systems available are expensive, complex, and require a database for tracking requirements. NUWC can't afford the overhead required by large projects and comprehensive systems when developing small to average-sized projects.

Project Overview

- Key Technical Challenges
 - developing a mechanism to track requirements within multiple documents throughout design, development, and testing
 - provide an adequate GUI for every type of user, including a novice user, to use the system
 - quickly providing the user with an auto-generated requirements traceability matrix for easy navigation between documents, which is developed solely by the team without influence from the customer
 - exporting the information into an interactive HTML or PDF using an XSLT processor and schema developed by the team

Technical Approach

After interviewing our customer, and agreeing on possible requirements, we realized we needed to provide a very light, easy-to-use platform, that is not currently available elsewhere. This platform needs to run in Internet Explorer running Windows 7, and our ultimate decision was to create a browser-based application capable of creating and modifying interactive Lifecycle documentation. Since this application runs in Internet Explorer, it will be:

- **cross-platform compatible** - can run on any operating system capable of running select browsers
- **require no install** - there is no need to create a formal installation, which may violate the Navy's security standards when creating registry keys
- **user-friendly** - allowing the user to easily navigate through documents and make modifications
- **provide form-like interface** - since our user is not familiar with XML, and how it works, we are hiding the XML code behind our graphical interface to avoid errors created by the user manually editing the XML

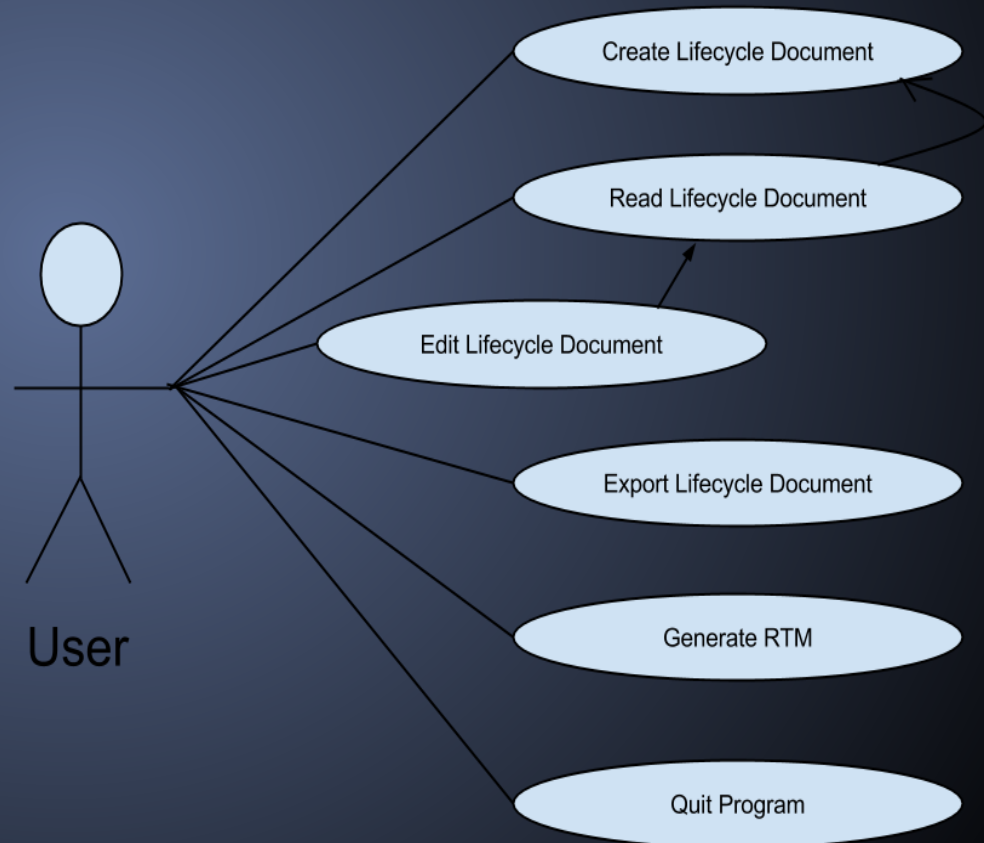
Software Development Process - SCRUM

Planning Sprint (9-13 thru 10-20)

ETC = 20 hours ATC = 15 hours

- Contact Customer
- Create Vision Document
- Start Modeling Use-cases
- Research new technology

Retrospective: This introductory sprint was generally to get the entire team onto the same page and working in unison. We realized the tools we were using were not up specifications required by the course, and other scrum tools very difficult and not very user friendly.



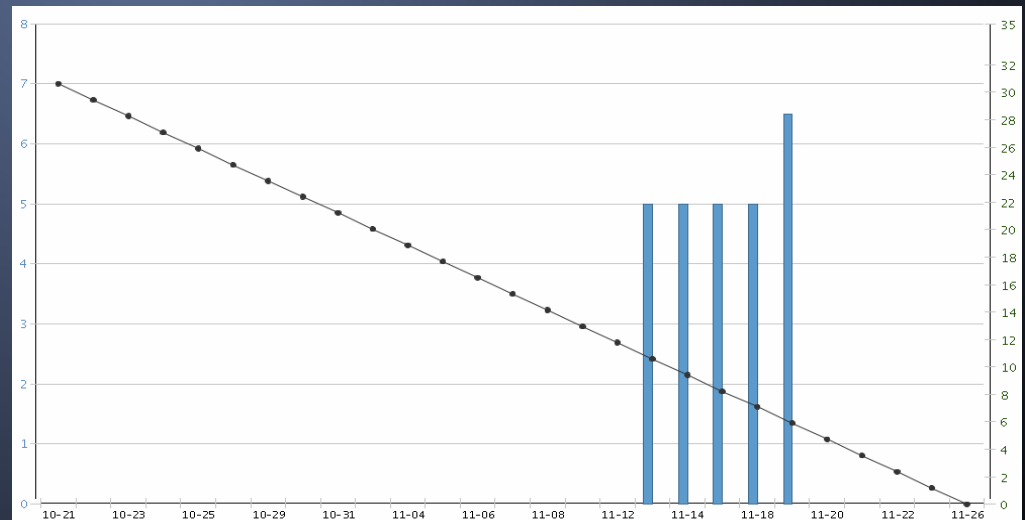
Software Development Process - SCRUM

Design Sprint (10-21 thru 11-27)

ETC = 10 hours ATC = 5 hours

- Update Vision Document
- Create Software Development Plan
- Create Software Requirements Specification

Retrospective: With a couple of planned meetings with our customer completed, we had a good idea of the project direction. Along with modeling, these meetings helped us a lot to prepare for development of the project. However, our scrum tools were still not up to par, but we finally decided to stick with Rally Development.



Software Development Process - SCRUM

Creation Sprint (11-28 thru 2-14)

ETC = 25 hours ATC = TBD

- Implement new technologies
- Create XML Schema
- Create Requirements Traceability Matrix
- Create Prototype

Retrospective: Our coders realized that having a much better knowledge of the programming languages would help better in this sprint. The models we created in earlier sprints worked well to define project direction, but should be a lot more comprehensive by the end of this sprint.



Software Development Process retrospective

- Interaction with customer

- Our first meeting on Sept. 26th, we met Michael in-person on 3rd floor Dion. We discussed his ideal project direction and ultimate goal. We learned that he needed a complete system to allow a user to create, and modify XML documentation, without prior knowledge of XML.
- The next two meetings with Michael during October in the library, were very informational in regards to how XML functions. Michael also provided an in-person demonstration of how an old and outdated system was used, and provided a presentation on XML, which gave us a more in-depth understanding of how the system was going to be used.
- The last meeting with Michael on Nov. 7th, we discussed the project progress. We also presented quite a few models, XML templates, and pencil drawn user interfaces with Michael to gauge our understanding and acquire more knowledge about the project goals.

Software Development Process retrospective

- **Jeremiah** - Team lead, Developer
 - accountable for the overall project completion as a lead developer and team manager
- **Peter** - Communications, Designer, Organizer
 - provides a contact point for our customer and helps coordinate team with organization of material in a timely manner
- **Kevin** - Quality Control, Documentation
 - creates and populates the required documentation in accordance with the team collaboration and project direction
- **Ari** - Developer, Designer
 - handles coding and visual display of the project, also works to assist the team lead with various tasks

Software Development Process retrospective

- **Miscommunication** in use of SCRUM tools at beginning of semester mislead the group into using multiple SCRUM tools. After using SCRUMDo and Trello, we finally decided to stick with Rally Software Development only because it provided time of completion in it's stories and illustrated a burndown graph for each iteration.
- **Sprint Iteration Burndown** is inaccurate due to the team utilizing multiple SCRUM tools before finally devoting to Rally.
- **Frequent team meetings**, and multiple meetings allowed the team to interact with the customer and gather a lot of information regarding the project. With each recurring meeting, the team learned more, and made new discoveries from customer interaction. Our NUWC advisor is very flexible when scheduling meetings here on campus.
- **Third Party IDE “Koding”**, initially was very buggy, but has since then greatly improved and is being utilized by the team to collaborate with code.
- **Google Drive** allows the groups to share all other documentation and material in a cloud for easy retrieval on campus and allows the team to work simultaneously in a particular document. This is used in combination with GitHub to store all the groups material.

Software Development Tools

- Describe all tools being used
 - Version and Source Code Control (and backup)
 - **Github** - web-based revision control system for software development projects that also provides a repository (ie. code)
 - **Google Drive** - file-storage and synchronization provided via a cloud storage which is excellent for the team to collaborate and work simultaneously on a document (ie. similar to Microsoft Office)
 - Standardized development environment (IDE)
 - **Koding** - online software development environment that allows developers to program in multiple languages and collaborate inside an internet browser without the need for a software development kit

Software Development Tools

- Scrum Tools

- Rally Software Development - an application lifecycle management software that provides a cloud-based solution to manage software development by providing an interface to plan, track, and automate the software development process

- List of 3rd party libraries

- jQuery - free and open source multi-browser library that was designed to simplify client-side scripting, making it one of the most widely used Javascript libraries
- XMLHttpRequest - offers an application program interface to be used with Javascript, this allows for simple interaction with XML documents. Incoming data can be used to alter the current document in the browser window without loading a new page

Demonstration

Prototype / GUI

Creating a Project

Init Create

display : "Insert filename"
display : "Available templates"

entry / create
do : create file on system
do : populate file with template
(Possible one line solution; "do : copy template with given filename")

Open State

O

Project Name :

Template :

Browse

Destination :

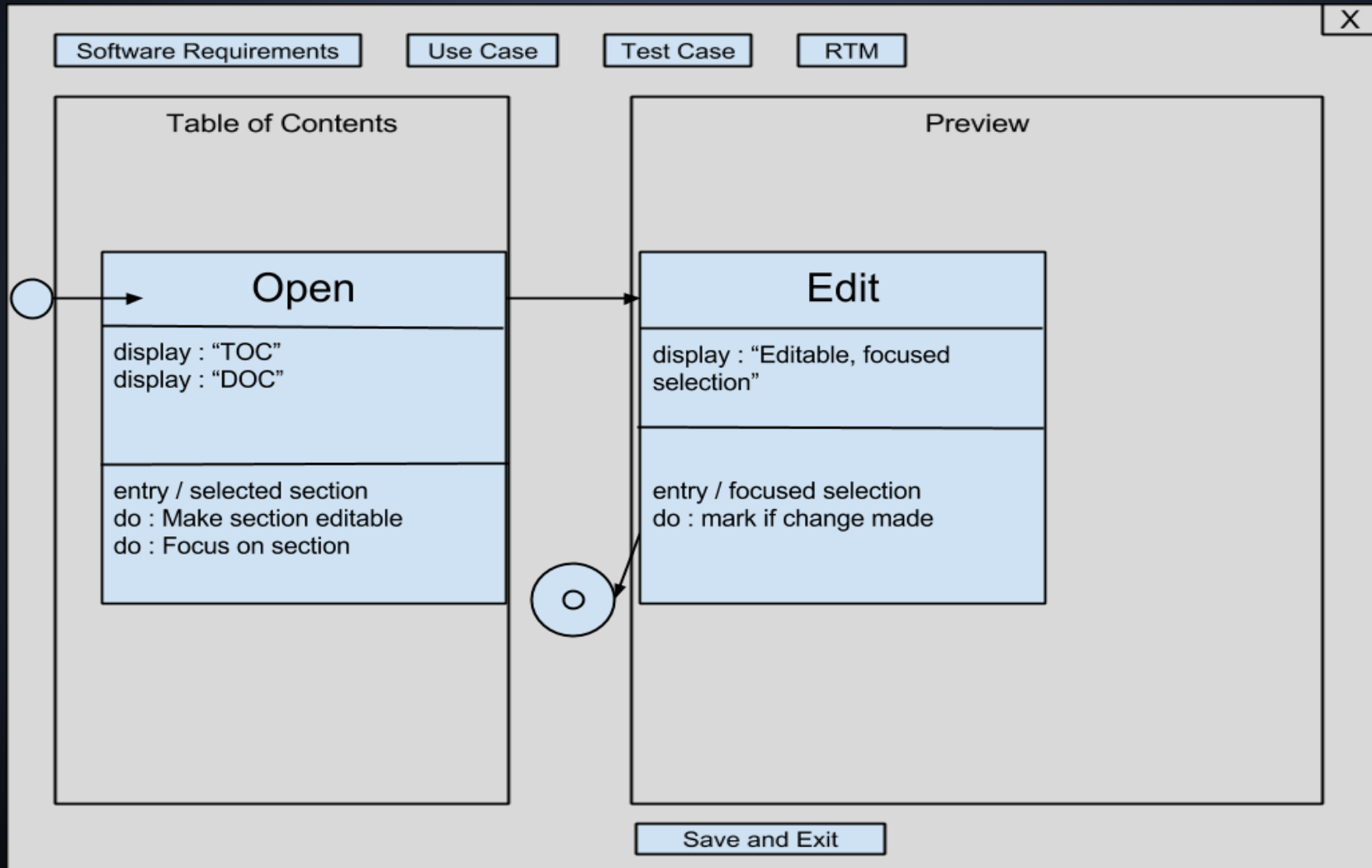
Browse

Cancel

Create

X

Editing a Project



Exporting a Project

