

## C868 – Software Capstone Project Summary

### Task 2 – Section A



Capstone Proposal Project Name:

Project Dashboard Simplify

Student Name:

Thuy Tien Nguyen

## Table of Contents

<b>Business Problem.....</b>	<b>3</b>
<b>The Customer.....</b>	<b>3</b>
<b>Business Case .....</b>	<b>3</b>
<b>Fulfillment .....</b>	<b>4</b>
<b>Existing Gaps .....</b>	<b>5</b>
<b>SDLC Methodology.....</b>	<b>5</b>
<b>1. Product Backlog Creation .....</b>	<b>7</b>
<b>2. Release Backlog .....</b>	<b>7</b>
<b>3. Sprint Backlog Creation.....</b>	<b>7</b>
<b>4. Working on Sprint and Scrum Meetings .....</b>	<b>7</b>
<b>5. Testing and Product Demonstration .....</b>	<b>8</b>
<b>6. Retrospective and Next Sprint Planning.....</b>	<b>8</b>
<b>Deliverables.....</b>	<b>8</b>
<b>Project Deliverables .....</b>	<b>8</b>
<b>Product Deliverables .....</b>	<b>9</b>
<b>Implementation.....</b>	<b>10</b>
<b>1.Concept.....</b>	<b>10</b>
<b>2.Inception.....</b>	<b>11</b>
<b>3.Iteration/Construction.....</b>	<b>11</b>
<b>4.Release .....</b>	<b>11</b>
<b>5.Production .....</b>	<b>11</b>
<b>6.Retirement.....</b>	<b>12</b>
<b>Validation and Verification.....</b>	<b>12</b>
<b>Environments and Costs.....</b>	<b>12</b>
<b>Programming Environment.....</b>	<b>12</b>
<b>Environment Costs .....</b>	<b>13</b>
<b>Human Resource Requirements.....</b>	<b>14</b>
<b>Project Timeline.....</b>	<b>15</b>
<b>Sources.....</b>	<b>16</b>

## Business Problem

### The Customer

Our customer is Creative Leaf Design and Build (CLDB), a small start-up home improvement company in San Diego, California. It provides home improvement and remodels services across San Diego County. Currently, the company has employed 100 staff members including full-time, and part-time employees together with other contractors and subcontractors. The company is expecting to be developed and grow because they continue to expand its business across San Diego County. The company's mission is to provide the best home improvement services to meet and exceed its customers' expectations.

### Business Case

Currently, the employees only use the paper-based planner to manage their monthly project which includes the project title/customer name, project leader, location, type, status, note, updated date, and estimate to complete date (ETC). When they get the new project, usually at the kick-off meeting the team members must write all information on their planner which is not efficient because they may lose their records, and hard to track whenever there is an update. In the future, more projects are coming, and the database will be larger and too complex to handle. Therefore, they need a simple, on-the-go, and mobile-friendly web-based dashboard application with a robust centralized database to help with their project management. Because the company is just a start-up business with little funding, they need a flexible budget and a simple application if possible; therefore, a software solution also needs to address this requirement in the delivery.

## Fulfillment

Project Simplify Dashboard is the web-based application dashboard backed by the MongoDB database to centralize all data from user input. MongoDB database is a non-relational document database that provides support for JSON-like storage. To access the dashboard, the users must register first, then they can use their credentials to log in. In the dashboard, the users can modify their credentials as they like when they navigate to their profile link.

The applications have three main navigations: project status home page, adding a new project page, and showing an all-projects page.

After logging in, on the homepage, the user can see an overall report of all projects like how many projects on working, finished, or canceled together with a line graph showing the number of projects for each month based on the estimated completion date (ETC). Also, users can opt to see a card view option showing the number of projects each month based on ETC.

There are 2 options for navigation to view all projects: the card list view and the table list view. Both options have action buttons for users to edit or delete a project they have created. Each project can be linked to a page for users to see all details of that project. On the project details page, users can have a chance to edit also.

For the navigation to add a new project, users can add a new project with these input prompts the project title or customer name, project leader or manager name, project location/address, project type, project status, project notes, and project estimate to complete date (ETC). This page is also used for editing a project when users click on the edit button.

The navigation bar is responsive and can be toggled to show or hide based on users' mobile devices because the team members constantly are on the go to the project locations and mostly access

the dashboard via their phones or company-provided tablets. Therefore, the dashboard should be mobile and user-friendly whenever possible.

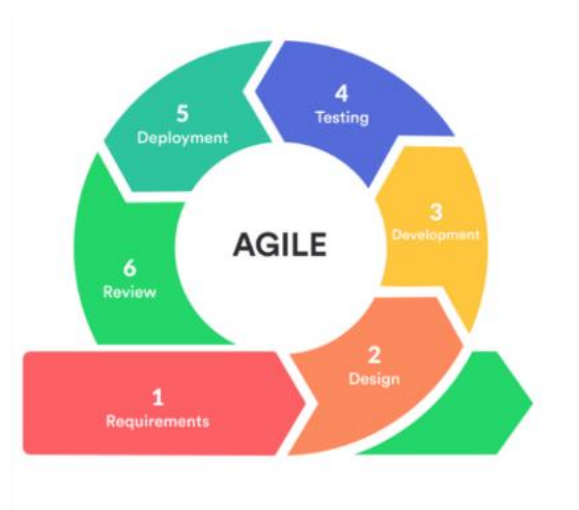
### Existing Gaps

The existing system is a manual and handwritten process using a paper-based planner. The team members always carry their planners to write down everything when they have a new task in a new location. This outdated method will be replaced with an electronic mobile device like a tablet which the company will provide to their team members after the Project Dashboard Simplify application is deployed officially.

With the assistance of a new web-based dashboard, the team members only need to carry their company-provided tablets or their own mobile devices to access their project dashboard and process information on the go without worrying about losing their records later. Whenever a project manager/leader needs a quick update or project status on a particular project they are working on, they can log in, keep track, search, or filter for information they need promptly.

### SDLC Methodology

Because this is a simple, responsive, and scalable application to start with a focus on user-friendly and mobile-friendly design, the application needs to be flexible to adapt to any change from customers' requirements or add-on features later, we will use Agile methodology for this approach. The agile method is based on continuous iterations of software that allow companies to release updates to users more frequently. Each new release serves as a base for the next one as presented in the figure below. For this Agile method, we will use Scrum workflow as follows:



### The Main Phases of SDLC Models

Source: <https://mlsdev.com/blog/agile-sdlc>

Scrum workflow is an iterative development process where development includes several phases. It is developed for teams who divide their work into goals, that can be completed within time-framed iterations called sprints, which last around 2- 4 weeks. The scrum team tracks the progress of the project in 15-minute stand-up meetings called daily Scrum. After sprint completion, the team holds a review meeting to demonstrate the work done and a retrospective to continually improve. The core roles involved in the Scrum process are Product Owner, Scrum Master, and Development Team as below workflow:



### Scrum Workflow Diagram

Source: <https://mindmajix.com/scrum-workflow>

## **1. Product Backlog Creation**

This is when the Stakeholders meet to decide the list of features that should be implemented and develop a product roadmap. In Scrum, features are known as user stories and are written from the end-user perspective. A Product owner decides which user stories or items make it into the product backlog.

## **2. Release Backlog**

After the product backlog is completed, the team together with the product owner decides how to group user stories into releases. Then, the development team estimates the time duration needed to complete each item. Once the release planning has been completed, the user stories are then selected for a sprint.

## **3. Sprint Backlog Creation**

A Sprint backlog is a set of product backlog items that must be delivered within a single sprint iteration. The duration of each Sprint lasts 2-4 weeks. Once the sprint backlog is determined, the team then divides each user story into a task. And then in each Sprint, the product is developed.

## **4. Working on Sprint and Scrum Meetings**

After the user stories for the current phase are selected, the development process begins. For tracking the current working process, daily scrums are held, and a task board is used with a description of tasks needed for implementation. Daily scrums or daily stand-ups are conducted by development teams with about 15 minutes timeframe to monitor the progress made towards the Sprint Goal and progress performed in the Sprint Backlog, to adjust the plan for the rest of the Sprint. This progress is tracked through a burndown chart with a day-by-day measure of the work that remains in a given sprint. By using

the chart, the speed of work is defined and concluded to determine the number of user stories for the next Sprint.

## **5. Testing and Product Demonstration**

If all the user stories are completed, the sprint backlog is also completed. After sprint completion, a sprint review is held, and a working application will be demonstrated for acceptance by the customers. Based on their feedback, stakeholders will decide on any further actions to take if any change.

## **6. Retrospective and Next Sprint Planning**

To plan for the next sprint, a retrospective meeting will be held for about 90 minutes at maximum. In this meeting, the scrum team can ask themselves these questions: What went well? What did not go well? What should be done differently? This meeting helps to coordinate continuous changes in the sprint. Based on the status and progress of the project, the whole cycle repeats until the project is completed.

# **Deliverables**

There are 2 types of deliverables associated with the Agile SDLC: project deliverables and product deliverables.

## **Project Deliverables**

Because our project is based on Agile methodology, these deliverables are also called agile scrum artifacts which include product backlog, sprint backlog, and increments as followings:



**a. Product Backlog**

The product backlog is a list of new features, enhancements, bug fixes, tasks, or work requirements needed to build a product. The product backlog can be updated on-demand as new information is available which is required by the product owner between sprint cycles.

**b. Sprint backlog**

The sprint backlog is a set of product backlog tasks that have been promoted to be developed during the next product increment. Sprint backlogs are created by the development teams to plan deliverables for future increments and detail the work required to create the increment. Sprint backlogs are created by selecting a task from the product backlog and breaking that task into smaller, actionable sprint items. The sprint backlog is updated during the sprint planning phase of the scrum. If a team does not meet an expected date to deliver all the sprint tasks, the remaining sprint tasks will be carried out in the next sprint.

**c. Product increment**

A product increment is the customer deliverables that were produced by completing product backlog tasks during a sprint. It also includes the increments of all previous sprints. There is always one increment for each sprint and an increment is decided during the scrum planning phase. An increment happens whether the team decides to release to the customer after passing the user-acceptance test for approval and the sign-off from the customer.

## Product Deliverables

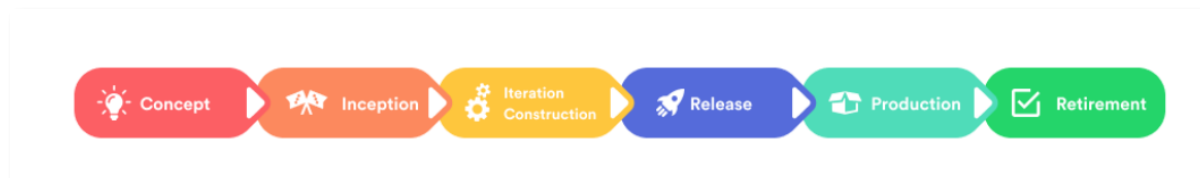
- ✓ Functional database with customized model and schemas using MongoDB third party.

## Project Dashboard Simplify

- ✓ Low fidelity wireframe on each sprint backlog to present navigation on the dashboard application.
- ✓ High-fidelity prototype on each sprint backlog to demonstrate the look and feel of the dashboard application.
- ✓ Fully developed web application with all functionality specified in the product backlog.
- ✓ Secured access to the application for all team members after they register.
- ✓ Deployment to Render platform as a secured and easy-to-manage cloud environment.
- ✓ Training support with a detailed user guide document.
- ✓ Maintenance support and further development if any update is required from the product owner.

## Implementation

The implementation of this application will be followed to a typical Agile workflow as following steps:



*The Main Phases of Agile Workflow*

Source: <https://mlsdev.com/blog/agile-sdlc>

### 1. Concept:

In the first stage of Agile SDLC, the Agile team makes the overall project evaluation and provides business analysis services. This is when the Agile team estimates how much time the project will require and identifies what resources might be needed later in development.

## **2. Inception:**

This is when the financial foundation occurs, and primary team members are identified to work on the project.

## **3. Iteration/Construction:**

This phase is divided into multiple timeframes which are called sprints, and there will be meetings every day to work on each sprint. During this phase, the Agile team starts to work on delivering the application. It is developed based on iteration requirements and continued feedback from customers. Because the core concept of the Agile method is welcoming changes and improvements, this phase is when the team will meet the customers to get feedback on each sprint before they can move on to the next one.

## **4. Release:**

This process includes QA testing, user-acceptance testing, documentation creation, and the released version of the application.

## **5. Production:**

After the application is deployed to the production environment, the software development team will train the team members from Creative Leaf Design and Build (CLDB) to register, log in, and how to use the application. After that step is completed, the team at CLDB should be able to start using the new application.

## 6. Retirement:

In this final phase, the development team stops supporting released software and notifies customers about it. However, the team can provide further service to keep up with any updates or add-on new features if there are requirements from customers in the future.

## Validation and Verification

In Scrum, testing is iterative based on planning sprints. The testing will include these activities:

- ✓ Contributing to User Stories based on the expected behavior of the System depicted as Test Cases
- ✓ Release Planning based on Test Effort and Defects
- ✓ Sprint Planning based on User Stories and Defects
- ✓ Sprint Execution with Continuous Testing
- ✓ Regression Testing after the completion of Sprint
- ✓ Reporting Test Results
- ✓ Automation Testing

## Environments and Costs

### Programming Environment

The application will be developed using the MERN stack which includes MongoDB for database management, NodeJS and ExpressJS framework for back-end development, and React framework for front-end development. Most of the code is written in JavaScript language with some HTML and CSS as supported component styles for front-end development. The database is backed by MongoDB database

## Project Dashboard Simplify

management third party. We will use MongoDB database manager for this application because it has a flexible data model to store unstructured data, and it provides full indexing support and replication with rich and intuitive APIs.

The application will be hosted in a shared cloud environment on the Render platform. According to Render, its site states *“Render is a unified cloud to build and run all your apps and websites with free TLS certificates, a global CDN, DDoS protection, private networks, and auto deploys from Git”*. The platform is a good choice to host our web application for easier management and maintenance.

## Environment Costs

The estimate for hosting on Render is based on the pricing plan. For a start-up company like our CLDB customer, the team plan is a good choice. According to Render’s pricing plan, the team plan is \$19 per user/month. Currently, two IT team members are working for CLDB to manage all tech-related tasks like database management and software management. So, the total cost for this team plan will be about \$38/month for hosting and about \$9.99/year for .com domain name registration.

Individual	Team	Organization	Enterprise
For hobbyists, students, and indie hackers.	For small teams and early-stage startups.	For larger teams with complex needs.	For ultimate power and customization.
<b>\$0</b> per user/month <u>+ COMPUTE COSTS</u>	<b>\$19</b> per user/month <u>+ COMPUTE COSTS</u>	<b>\$29</b> per user/month <u>+ COMPUTE COSTS</u>	<b>Custom Pricing</b>

Pricing Plan on Render

Source: <https://render.com/pricing>

## Project Dashboard Simplify

For database management service using Mongo DB vendor, as a start-up company a shared cluster “pay-as-you-go” plan is a good fit. The estimated cost for the base price is starting from \$9/month if CLDB chooses this plan with cluster tier M2 for 2 GB storage. In the future, when CLDB enlarges its business, it can upgrade to a dedicated plan easily.

The screenshot displays MongoDB's pricing structure. On the left, three main plans are highlighted: Serverless (starting at \$0.10/million reads), Dedicated (starting at \$57/month), and Shared (starting at \$0/month). Each plan includes a 'Sign Up' button and a brief description of its use case. To the right, a detailed 'Shared Cluster' table provides specific pricing for different cluster tiers (M0, M2, M5) based on storage, RAM, and vCPUs. The table also lists the base price for each tier: M0 is free forever, M2 is \$9/month, and M5 is \$25/month.

Cluster Tier	Storage	RAM	vCPUs	Base Price
M0	512 MB	Shared	Shared	Free forever
M2	2 GB	Shared	Shared	\$9/mo
M5	5 GB	Shared	Shared	\$25/mo

Pricing Plan on MongoDB

Source: <https://www.mongodb.com/pricing>

## Human Resource Requirements

The project requires a project manager (Scrum master), a UX/UI designer, one full-stack software engineer, and one QA specialist. For this human resource planning, a project manager will be hired part-time as Scrum Master to manage the whole Agile SDLC, the estimate for this project is about 4 weeks and this role will need about 20 hours/per week. The Development Team includes a part-time UX/UI designer, a full-time software developer, and a part-time QA specialist. The total cost for human resources for this project is estimated in the following table:

Resource	Rate * Time	Total
Project Manager (Scrum Master)	\$80/h * 80h (4 weeks, parttime – 20h/week)	\$6,400
UX/UI Designer	\$40/h * 80h (4 weeks, parttime – 20h/week)	\$3,200

Software Developer	\$100/h * 160h (4 weeks, fulltime – 40h/week)	\$16,000
QA Specialist	\$60/h * 80h (4 weeks, parttime – 20h/week)	\$4,800
<b>Total HR Cost</b>		<b>\$30,400</b>

## Project Timeline

The whole project is estimated to finish in around 1 month or 4 weeks and it will be determined by our customer as the Product Owner from Creative Leaf Design and Build (CLDB) based on their sign-off acceptance:

Phase	Milestone/Task	Deliverable	Description	Dates
<b>Concept</b>	Initial meeting with CLDB to understand project requirements	Project duration and required resources	Project evaluation and business analysis	03/01/2023-03/02/2023
<b>Inception</b>	Overall project planning	Project budget estimate and Development Team identified	Financial foundation occurs, and primary team members are identified to work on the project.	03/03/2023-03/04/2023
<b>Iteration/Construction</b>	Project execution starts to work on delivering the application	Product backlog, sprint backlog, and product increment	There are 2 sprints for this project, each will last about 10 days. The project team meets every day to work on each sprint, and there are several meetings with the product owner to get feedback on each	03/05/2023-03/26/2023

			sprint before they can move on to the next one	
<b>Release</b>	Project monitoring and controlling	Documentation, centralized database, and full-stack web application	QA testing, user-acceptance testing, documentation creation, and the first released version of the application. The timeline is paralleled with the iteration phase.	03/05/2023-03/26/2023
<b>Production</b>	Prepare for deployment	Product-ready application, user guide, and training support	Set up Render environment for deployment and live hosting	03/27/2023-03/31/2023
<b>Retirement</b>	Project closure	Lifetime maintenance/support service	Continuous support if any updates or requests from CLDB	Based on requirements from the customer/product owner

## Sources

MLSDev. *Agile SDLC*. MLSDev Blog, <https://mlsdev.com/blog/agile-sdlc>, 9 February 2023.

MindMajix. *Scrum Workflow*. MindMajix, <https://mindmajix.com/scrum-workflow>, 9 February 2023.

Render. *Pricing*. Render, <https://render.com/pricing>, 9 February 2023.

MongoDB. *Pricing*. MongoDB, <https://www.mongodb.com/pricing>, 9 February 2023.