Department of Computer Science & Engineering
The University of Texas at Arlington

Project Charter
CSE 4316: Senior Design I
Spring 2023

ERS
Equipment Reliability Strategies

Matthew Luu
Nicholas Nguyen
Pablo Sanchez
Hoang Long Quan Nguyen
Viraj Vipinbhai Sabhaya

Revision History

Revision	Date	Author(s)	Description
0.1	2.20.2023	ML, NN, PS, HN, VS	document creation
0.2	5.10.2023	NN, PS, HN, VS	Version 2

Contents

List of Figures

1 Problem Statement

Equipment breakdown can cause delays in the production system. A vital machine breakdown may also cause the whole production line or the whole factory to be shut down completely, which can cost manufacturers hundreds of thousands of dollars per day. Moreover, it also can cause health, safety, and environmental accidents as well. Therefore, there is a need for an application that helps organizations to manage their equipment and prevent those issues. Becht developed the "Equipment Reliability Program" a long time ago to help its customers meet their needs, but this application is outdated with old UI/UX, a lot of redundant fields, etc.

2 Methodology

According to research, around ninety percent of machine breakdown cases can be prevented with a good preventive maintenance plan. This is why Equipment Reliability Strategies Application was built for. This application will be a very user-friendly web-based application, using the latest technologies for both front-end and back-end, targeting organizations that own or operate process facilities across many industries. By using this application, the end-users, such as site managers, engineers, and inspection and maintenance staff will be able to add equipment, equipment history, strategies, and manage their equipment correctly according to those strategies, etc, which keeps equipment operating efficiently, minimize health, safety, and environmental accidents, and helps to avoid large and costly repairs down the road.

3 Value Proposition

Becht's current "Equipment Reliability Program" was built a long time ago, and is outdated with old UI/UX, a lot of redundant fields, etc. The Equipment Reliability Strategies Application will be a brand new application, built from scratch, using technologies currently and successfully being used by Becht. It will solve the same problems as the "Equipment Reliability Program", but with a better modern UI/UX, a new design approach, complex data hierarchy, and added functionalities.

4 Development Milestones

This is a list of milestones and completion dates for this project:

- Project Charter first draft - February 2023
- System Requirements Specification - March 2023
- Architectural Design Specification - April 2023
- Demonstration of UI mockups - April 2023
- Demonstration of ER Diagram - April 2023
- Detailed Design Specification - June 2023
- Demonstration of front-end features - June 2023
- Demonstration of back-end features - June 2023
- CoE Innovation Day poster presentation - June 2023
- Demonstration of front-end and back-end integration - July 2023
- Demonstration of software testing - July 2023
- Final Project Demonstration - August 2023

5 Background

Across most industries, millions of dollars are spent on equipment failure and maintenance. In addition, these failures may also cause related health, safety, and environmental accidents that must be addressed. That is the main reason why Becht developed the "Equipment Reliability Program" Application and use it to help organizations identify the causes of these equipment failures and apply strategies to void those failures from occurring again in the future. By using the service from Becht, organizations can further maximize their asset return on investment. However, as mentioned above, the "Equipment Reliability Program" Application is outdated and needs to be completely redesigned. Therefore, Becht sponsors this project, and our team will work to develop a brand new user-friendly web application, using technologies currently and successfully being used by Becht. It will solve the same problems as the "Equipment Reliability Program" Application, but with a better modern UI/UX, a new design approach, complex data hierarchy, and added functionalities.

6 Related Work

There are many currently commercially available equipment maintenance strategies developed by different organizations. Some examples include Arms Application, which provides its customers with training in equipment reliability, numerous tools involving equipment life cycle analyses, and other tools to aid in equipment reliability [1]. Lifetime Reliability Solutions Application and their courses on planning and scheduling for maintenance management [2]. TRACTIAN system of condition monitoring application monitors machines and equipment pieces around the clock, analyzing the data and providing practical insights into machine health. This way users can protect thier production line assets as well as the utility and balance of all plant equipment. It empowers users to detect failures early on with high accuracy and monitor automatically the more important KPIs for your operation, such as MTBF, MTTR, reliability and others. Everything from minute one after setting up thanks to its Plug and Play technology [5]. Fiix Application is a preventive maintenance software that helps teams run equipment maintenance, connect to business systems and make data-driven decisions [3]. Asset Essentials Application helps to save time and money through more efficient maintenance management and operations, reduce maintenance costs, empower service teams and community, and optimize resourcing and communication [4]. These organizations and their web applications fulfill similar goals to our product. However, Becht wants to have their own application that meets its own needs, utilize its powerful default strategies, and resources, and also works best for its business. That's why Becht sponsors us to develop this Equipment Reliability Strategies Application.

7 System Overview

The Equipment Reliability Strategies Application consists of four layers, from top to bottom, which are the Web, Core, Model, and Database layers. These architectural "layers" are the top-level logical view, or an abstraction, of the design. Those four layers have related functionality associated with each other as shown in Figure 1 below.

7.1 Web layer

The Web layer is the outermost layer, which runs on the user's computer. It is a space where the users interact directly with this web application to complete business tasks. The Web layer will receive input data from the users, send requests to the server, receive responses from the server, and display the data from those responses to the users in a way that they can understand.

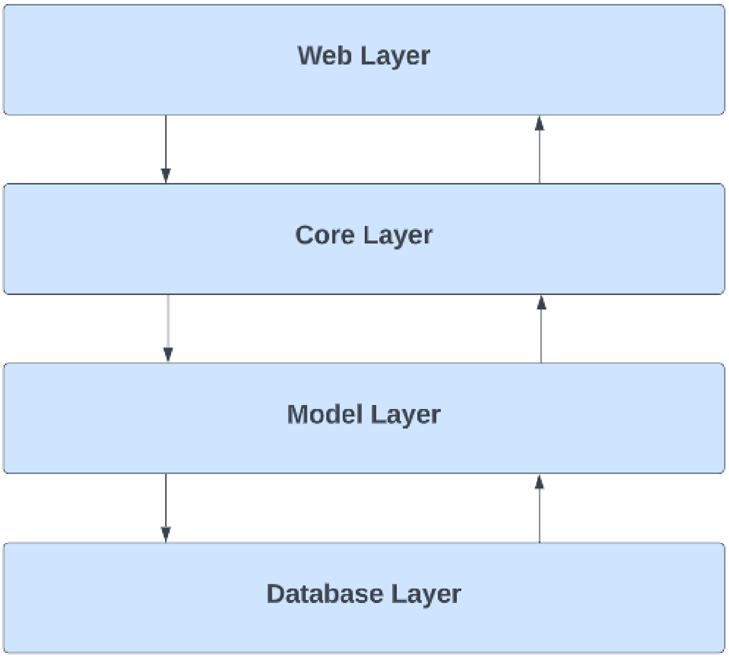

Figure 1: Architectural layer diagram

7.2 Core layer

The Core layer contains the business logic of the application, including the security, access, and authentication tasks. This layer is responsible for performing specific business rules associated with the user requests from the Web layer.

7.3 Model layer

The Model layer is where the database turns into objects or a client-side representation of an object. Its main functions are connecting/accessing the database, retrieving requested data, and sending it back to the Core layer.

7.4 Database layer

The Database layer is where all the database schema and application data are stored securely on the server side. Its functions are receiving requests from the Model layer, and sending responses including data back to it accordingly.

8 Roles & Responsibilities

The key stakeholders and resources on the Becht, in the format of Name: Title, Project Role(s).

- Abby King: Vice President of Plant Services, Product Sponsor
- Lakeshia Taylor: Global Reliability Lead, Product Owner
- Mark Fronek: Digital Solutions Manager, Business Analyst, and Capstone Sponsor
- Agustin Ruiz: Talent Development Lead, Primary Technical Contact
- Jon Yutzy: Technical Excellence Lead, Solution Architect

- Ryan Vallette: UI/UX Designer, User Interview/Testing
- Chris Agbanyo: Jr. Systems Administrator, Cloud resource management
- Ameze Goubadia: SCRUM Trainer, Agile coaching

The task of creating this web-based application has been given to Team ERS. The team consists of:

- Matthew Luu
- Nicholas Nguyen
- Pablo Sanchez
- Hoang Long Quan Nguyen
- Viraj Vipinbhai Sabyhaya

Delegation of responsibilities for each team member's individual web applications component will take place during a regularly scheduled meeting. Each team member will be accountable for learning more about their particular assignment and taking into account their own strengths and skills sets.

The product owner and scrum master for the project will be changed periodically to improve the awareness and responsibility of all team members and the efficiency of the project.

9 Cost Proposal

For cost, the money will mainly come from the CSE department and Becht if we need to pay for anything. However, Becht is already sharing many of the technologies that will be used for the project and so there doesn't seem to be a need to spend a lot of money for the project.

9.1 Preliminary Budget

Components	**Amount**	**Source**
Azure Cloud Service	$0	Becht
Lucid Account	$0	Becht
Pluralsight Account	$0	Becht

Table 1: Overview of budget

9.2 Current & Pending Support

The funding for the project primarily comes from the CSE department's default amount of $800, which is shown in table 2 below.

Source	Amount
CSE department	$800

Table 2: Current & pending support

10 Facilities & Equipment

For facilities we will to find meeting areas to use when meeting with team members from Becht which must fit at least 6 people, 5 from our team and at least 1 member from Becht, have internet access to allow for other Becht members that can't make it be able to interact via a Microsoft Teams meeting, and have a projector or TV so things can be presented to everyone in the meeting. For the project it self we would need a computer and to get a study room from the library to be able to easily collaborate with each other in person to help one another. However, outside of that for the project we don't need any other external facilities or equipment since it is mainly a digital web application that can be tested through data within a computer. Possible areas for in person meetings with Becht is with the CSE department requested through our professor, downtown Arlington Library if large enough, UTA Central Library if we can reserve an area large enough that also has a TV. For computer if one of us is without one we could go use one in the UTA Central Library which is open to all UTA students for free and they have many that are open to use. Study rooms to work together in can be easily reserved within the Central Library just have to make sure to do it ahead of time or could also reserve a study room in the Arlington downtown library if the one in UTA is full.

11 Assumptions

The following list contains critical assumptions related to the implementation and testing of the project.

- Data to test the application will be given by Becht
- Becht will provide UI assets to be cohesive with the rest of their designs
- Becht will provide feedback to what we have
- Azure will be used to host the code

12 Constraints

The following list contains key constraints related to the implementation and testing of the project.

- Must use MSSQL for the DB
- Must use the group Azure from Becht to host the code
- Must maintain similarity with the other webpages
- Total development costs must not exceed $800
- Must be completed before the end of SD2 in the summer August

13 Risks

This section contains a list of the most critical risks related to this project, including the probability of occurrence, size of loss in days, and risk exposure in days. Mitigation strategies will be discussed in future planning sessions.

Risk description	**Probability**	**Loss (days)**	**Exposure (days)**
Scheduling conflicts between team members	0.50	20	10
Team members are inexperienced in new technologies	0.20	14	2.8
Team members are busy with other courses	0.30	9	2.7

Table 3: Overview of highest exposure project risks

14 Documentation & Reporting

14.1 Major Documentation Deliverables

These deliverables are major grade components of the course. Completing these documents should generally be the sprint goal during the applicable sprint period. Refer to current and previous course syllabi and schedules to estimate the due dates of these items. Remove this explanatory paragraph from your draft, but leave the heading.

14.1.1 Project Charter

The team will be in charge of maintaining the Project Charter, and it will be updated on account of any modifications made to the project by team members. In order to make it simple to modify the project charter, it will be maintained using a single shared account on Overleaf.

There may be revisions made to the charter as a result of the project's objectives and constraints from the initial draft.

The first draft will be completed as soon as we get all the information we need to complete the Project Charter, and the final product will be handed in the second week of August.

14.1.2 System Requirements Specification

The System Requirements Specifications will contain important details about the development, quality assurance, operation, and maintenance. This document, which should be kept up-to-date on Overleaf so that all team members can access it, keeps everyone informed about the life cycle of the product. it will be used to inform the team of features from the first SRS that needs to be discontinued.

Before the project's first iteration, the initial version will be delivered, and the final version will be delivered before the week of finals.

14.1.3 Architectural Design Specification

Architectural Design Specification is a technical document that describes how a software system is to be developed to achieve the goals described in the requirements. This document must be frequently written, updated, and maintained on Overleaf. However, as the objectives of this project will remain the same, there will not be many revisions to this document.

The first draft will be completed following the system requirement definition, and the finished product will be handed in at the end of the spring semester.

14.1.4 Detailed Design Specification

The functional requirements stated in the Functional Specification are developed and implemented according to the thorough design specification. This document, which describes the functional specification of the software, will be prepared once the initial version of the web application is developed. In order to make it clear to users what each button field does, it will be maintained by the team on Overleaf and updated each time a function is added after its creation.

The initial version will be provided following the release of the first edition of the web application, and the final version will be delivered in August.

14.2 Recurring Sprint Items

The following items will be documented and maintained during each individual sprint.

14.2.1 Product Backlog

All of the requirements from Becht will be gathered, and after that, they will be added from the SRS to the product backlog. We will use the agile method to order the requirements because we are building this product from scratch. We'll choose which requirement will be put into practice first at one of our initial group meetings. We will use Overleaf to maintain and share the backlog information with other teammates and stakeholders.

14.2.2 Sprint Planning

We will follow the syllabus to manage our sprint duration. For Senior Design 1 and Senior Design 2, we initially plan 4 sprints each. After more discussion with the team members, this may change in the future if necessary.

14.2.3 Sprint Goal

The Scrum master will select the Sprint Goal. We will involve our client at each stage of the project as we move forward because we are adopting an agile methodology for this project.

14.2.4 Sprint Backlog

The product backlog items' selection in the sprint backlog will be decided by the team lead. The software will be used to maintain the backlog.

14.2.5 Task Breakdown

For task breakdown, we'll stick to a straightforward strategy. The team leader will assign each team member specific tasks based on their expertise in each field. To record the amount of time we spend on various tasks, we will utilize an engineering notebook.

14.2.6 Sprint Burn Down Charts

The creation of the burn-down charts for each sprint will be the responsibility of the team lead. The team leader can retrieve the overall amount of work required to produce a final burn-down chart using the time recorded in the Engineering notebook. The burn-down charts will be represented graphically.

14.2.7 Sprint Retrospective

After each team meeting about the sprints the scrum master and the team will manage the sprint retrospective. The collective task will be recorded collectively, while the individual task will be recorded individually. The assignment is due at the end of each sprint.

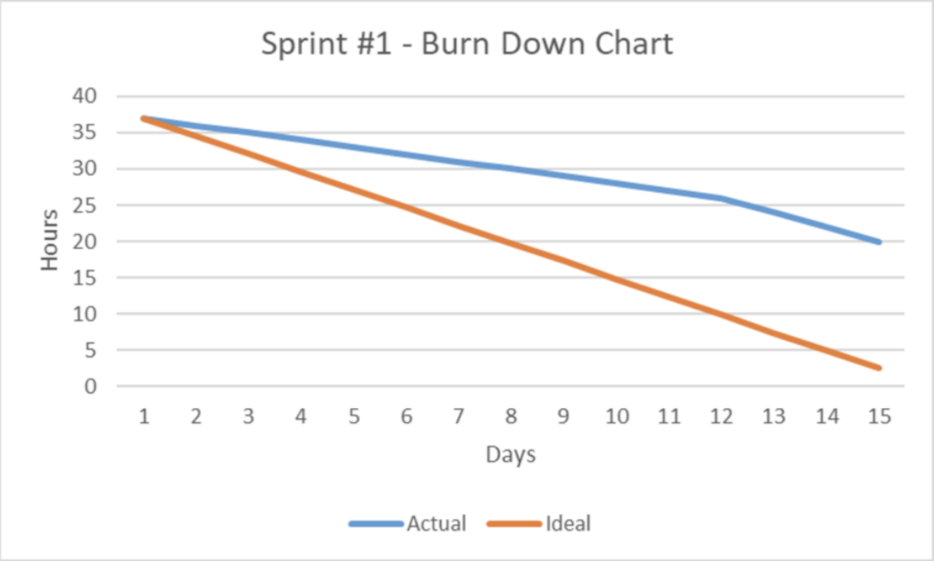

Figure 2: Example sprint burn down chart

14.2.8 Individual Status Reports

Each person will provide a status report once the task they were given has been finished. Each individual status report that team members provide will include the specific components of the task that they covered.

14.2.9 Engineering Notebooks

Once we complete the project-specific work, we will update the engineering notebook. The sort of task that is being worked on will determine how long it takes to update the time for each task. The team leader's task assignment will also be recorded in the notebook for future use. We will not need a witness signature for our ENB.

14.3 Closeout Materials

14.3.1 System Prototype

The final system prototype will feature a web application. We will demonstrate in class in August through a laptop.

14.3.2 Project Poster

The poster will be 3 x 4 feet (36" x 48") in size and incorporate the project's vision, mission, architectural design diagram, essential needs, and upcoming work, in addition to web application screenshots. On August 20, 2023, the poster will be delivered.

14.3.3 Web Page

The web page's content will be identical to that of the poster. It will be published on the CSE blog and the UTA website. Delivery of this is scheduled for August 2023. It will be revised as the project progresses.

14.3.4 Demo Video

The user will be seen using all of the application's functions in the demo video. It will display distinct functionality. The video, which will last around ten minutes, will demonstrate how the app can be used.

14.3.5 Source Code

We'll use Git to maintain our source code. We are putting in place a distributed version control system utilizing Git. To keep the application's confidentiality from competitors, the source code will not be made available to the customer. The general public will not have access to the project's source code.

14.3.6 Source Code Documentation

The final documentation will be PDF, and we'll be using Doxygen or Javadocs. to aid the developers in understanding the purpose of our code and the reasoning behind its lines of code, we will provide an explanation using comments.

14.3.7 Hardware Schematics

N/A

14.3.8 CAD files

N/A

14.3.9 Installation Scripts

The sponsor will deploy the software. They will also provide the scripts, install programs, and any tools to improve the process.

14.3.10 User Manual

The sponsor will provide the user manual for the users to learn how to maneuver around the web application.

References

[1] Improve Equipment Reliability.

[2] Online Training Courses. Technical report.

[3] Preventive maintenance software features and benefits.

[4] Smarter Asset Management.

[5] What your machines need is a daily check-up.