5-2016

Development of Web Application for Monitoring Rail Track Quality

Prudhvi Krishna Kodali
St. Cloud State University

Follow this and additional works at: https://repository.stcloudstate.edu/mme_etds

Recommended Citation
https://repository.stcloudstate.edu/mme_etds/30

This Starred Paper is brought to you for free and open access by the Department of Mechanical and Manufacturing Engineering at theRepository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Mechanical and Manufacturing Engineering by an authorized administrator of theRepository at St. Cloud State. For more information, please contact rwexelbaum@stcloudstate.edu.
Development of Web Application for Monitoring Rail Track Quality

by

Prudhvi Krishna Kodali

A Starred Paper
Submitted to the Graduate Faculty of
St. Cloud State University
in Partial Fulfillment of the Requirements
for the Degree
Master of Engineering Management

May, 2016

Starred Paper Committee:
Ben Baliga, Chairperson
Hiral Shah
Balasubramanian Kasi
Abstract

All the rail industries in the present world spend lot of money and resources on track inspections. There are various modern technologies in the modern world for the rail track inspections but neither completely operable at the ground level nor economical for the industries. There is no specific tool or application available in the present market which intakes all the track related data, keep track of the inspections and notify users the schedule of inspections. All this process depends upon the quality of data being collected from the tracks, defining objective from the collected data and achieving the results. For this purpose in this project objectives for the data collection has been given prominent importance and it is done in two ways. Validations are performed before the data is stored into the database.

In order to analyze, define and project the data into useful form a robust application is needed which responds to user request dynamically. Our mission here was to understand the user requirements to keep track of the rail track data and automate the process of scheduling inspections and notifying the users.

Therefore this project mainly focuses on developing the robust web application which checks the quality data being collected, properly validated and interact with users to help them in scheduling and maintaining inspections. This project helps management to bring legacy systems to bring under one roof and also to expand the existing application in the future according to the changing needs.
Acknowledgement

I am grateful to my advisor, Director of EM program Dr. Ben Baliga for the immense guidance provided by him throughout the course of this project. By sharing his thoughts and suggestions he guided me to complete this project successfully. Dr. Baliga spent his valuable time to read and understand this paper. Dr. Baliga’s suggestions and the recommendations were very helpful for me to design and implement this project in a more effective manner.

I also convey my sincere thanks to Dr. Hiral Shah, Dr. Gary and Dr. Balsy Kasi for their support and suggestions during the entire period of this project. I am grateful to Engineering Management program and St. Cloud State University for providing me such wonderful resources. And finally I would like to thank my family and all my well-wishers who supported me throughout my master’s program.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Table</td>
<td>6</td>
</tr>
<tr>
<td>List of Figures</td>
<td>7</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>I. Introduction</td>
<td>8</td>
</tr>
<tr>
<td>Problem Statement</td>
<td>9</td>
</tr>
<tr>
<td>Nature and Significance of the Problem</td>
<td>9</td>
</tr>
<tr>
<td>Objective of the Project</td>
<td>10</td>
</tr>
<tr>
<td>Project Questions</td>
<td>10</td>
</tr>
<tr>
<td>Assumptions</td>
<td>11</td>
</tr>
<tr>
<td>Definition of Terms</td>
<td>11</td>
</tr>
<tr>
<td>Summary</td>
<td>12</td>
</tr>
<tr>
<td>II. Background and Review of the Literature</td>
<td>13</td>
</tr>
<tr>
<td>Introduction</td>
<td>13</td>
</tr>
<tr>
<td>Background Related to the Problem</td>
<td>13</td>
</tr>
<tr>
<td>Literature Related to the Problem</td>
<td>15</td>
</tr>
<tr>
<td>Literature Related to the Methodology</td>
<td>19</td>
</tr>
<tr>
<td>Summary</td>
<td>21</td>
</tr>
<tr>
<td>III. Methodology</td>
<td>22</td>
</tr>
<tr>
<td>Introduction</td>
<td>22</td>
</tr>
<tr>
<td>Design of Study</td>
<td>22</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Data Collection</td>
<td>32</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>35</td>
</tr>
<tr>
<td>Budget</td>
<td>36</td>
</tr>
<tr>
<td>Timeline</td>
<td>36</td>
</tr>
<tr>
<td>Summary</td>
<td>37</td>
</tr>
<tr>
<td>IV. Data Presentation and Analysis</td>
<td>38</td>
</tr>
<tr>
<td>Introduction</td>
<td>38</td>
</tr>
<tr>
<td>Data Presentation</td>
<td>38</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>40</td>
</tr>
<tr>
<td>Summary</td>
<td>45</td>
</tr>
<tr>
<td>V. Results, Conclusions, Inferences, and Recommendations</td>
<td>46</td>
</tr>
<tr>
<td>Introduction</td>
<td>46</td>
</tr>
<tr>
<td>Results</td>
<td>46</td>
</tr>
<tr>
<td>Project Questions</td>
<td>47</td>
</tr>
<tr>
<td>Conclusion</td>
<td>50</td>
</tr>
<tr>
<td>Recommendations</td>
<td>51</td>
</tr>
<tr>
<td>References</td>
<td>53</td>
</tr>
</tbody>
</table>
# List of Table

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. List of Technologies Used</td>
<td>25</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Spring MVC Framework Architecture</td>
<td>16</td>
</tr>
<tr>
<td>2.</td>
<td>Hibernate ORM Framework</td>
<td>17</td>
</tr>
<tr>
<td>3.</td>
<td>Spring Database Integration</td>
<td>18</td>
</tr>
<tr>
<td>4.</td>
<td>Lean Development Flow</td>
<td>20</td>
</tr>
<tr>
<td>5.</td>
<td>SDLC</td>
<td>24</td>
</tr>
<tr>
<td>6.</td>
<td>Work Breakdown Structure</td>
<td>27</td>
</tr>
<tr>
<td>8.</td>
<td>Defect Raised in Bugzilla</td>
<td>31</td>
</tr>
<tr>
<td>9.</td>
<td>Sheet for Entering the Data</td>
<td>39</td>
</tr>
<tr>
<td>10.</td>
<td>Data Display in Application</td>
<td>40</td>
</tr>
<tr>
<td>11.</td>
<td>Application Data Defects</td>
<td>41</td>
</tr>
<tr>
<td>12.</td>
<td>Defect Arrival and Kill Rate</td>
<td>42</td>
</tr>
<tr>
<td>13.</td>
<td>Defects in Code</td>
<td>43</td>
</tr>
<tr>
<td>14.</td>
<td>Total Number of Errors from the Application Produced Reports</td>
<td>44</td>
</tr>
<tr>
<td>15.</td>
<td>Defects Identified in Different Months</td>
<td>45</td>
</tr>
</tbody>
</table>
Chapter I: Introduction

ENSCO, Inc., and its wholly owned subsidiaries represent a $90 million technology enterprise, headquartered in the Washington, DC area. For more than four decades, the ENSCO group of companies has been providing ingenious engineering, science and advanced technology solutions that guarantee mission success, safety and security to governments and private industries around the globe. It operates in the defense, transportation, aerospace and intelligence sectors.

Track Management is one of the major projects that is currently in progress for the Australian coal mining company. Train that carries the coal from the mine to the sea coast travels on a track that passes through hills, desert and several other challenging areas. A tool or application is required to manage the track inspection, maintain its assets, and monitor the changes. ENSCO’s Data Management System addresses the challenges of determining where to focus inspections, how to plan for maintenance, and how to ensure the effectiveness of these decisions. **RTCM (Rail Track Condition Monitor)** addresses these issues with a simple to use, low-cost solution that provides senior railroad management, engineering and maintenance-of-way personnel with a window into the condition, infrastructure and planning that takes place on a railroad.

It simplifies the enterprise-wide storage, analysis and dissemination of track infrastructure, maintenance and inspection data. This project ensures the people working on field to focus only on the areas required and to work on their tasks as per schedule and gets updated on the track issues.
Problem Statement

Operators working on the track need the updated data about the track and its required inspections, features, defects, quality and maintenance history. They need this data not just in the form of numbers or calculations. They need it in the form of tables, charts, graphs where they can interpret the track in its present condition and perform their duties. They require an application which tells them to perform inspections at the specific time and at specific location. They require a tool which presents them with all the required data about the track and gives them options to choose what are the locations and specific times where they need to perform inspections or repairs. So in order to have this tool present all its data in the form of user friendly format initial data regarding the track should be collected. This is basically done in two ways. One is operator working on the field entering the data into system and other one is V/TI, an automated vehicle which collects the data and store it in the database.

Nature and Significance of the Problem

There are no such tools exist at present in the Rail Track Management history which intensively concentrates on the rail track and closely monitor it. So there exist the challenges of determining where to focus inspections, how to plan for maintenance, and how to ensure the effectiveness of these decisions.

RTCM integrates common railroad data, such as track geometry and rail wear defect rates, track quality index, tonnage, track speed, curvature, infrastructure and program maintenance history, into consolidated graphical and reporting views with an
intuitive interface that requires minimal end-user training. The resulting product provides a means to rapidly access inspection data, brush charts, track charts, maps, reports and multi-media information, such as photos of the right of way or track roadbed. This solution significantly reduces the time and labor required to access information commonly used to plan inspections and maintenance, investigate track-related accidents, and generate reports or data exports that can be taken into the field.

**Objective of the Project**

The main objective of the project is to develop this RTCM web application which can be used by the track operators while they are on the track. Develop this web application in such a way that this can be used remotely on the track where they could be no internet and all the data should be fetched from offline database and present it to the operators working. Present the data in very presentable way and show them in the maps where exactly the work is to be performed on tracks. Develop a solution that can significantly reduce the time and labor required to access information commonly used to plan inspections and maintenance, investigate track-related accidents, and generate reports or data exports that can be taken into the field.

**Project Questions**

After successfully accomplishing the project, the following potential questions could be answered based on the results obtained from the project:

- What are the ways to identify the defects in tracks?
- Where to perform schedule of inspections for specific tracks at specific location?
- Was RTCM providing correct information about the data and its assets, if yes were there any failures in identifying defects in tracks?
- Was RTCM really helpful in providing data on field where there is no internet and also providing them with correct location?
- Does RTCM really reduced the labor required to access information commonly used to analyze track conditions?

Assumptions

- All the data collected is updated timely through nightly batch jobs and updates the database in all environments.
- Data collected through various means is validated before it is stored into a database.
- Generic programming language has been used to develop the code which can be run on any platform.
- Data collected through V/TI has all the required fields and data collected manually has been validated.

Definition of Terms

**RTCM:** RTCM stands for Rail Track Condition Monitor, name for the application being developed. It indicates the application being developed is used to monitor the condition of track without much manual involvement. Java is the programming language being used to develop this application.
MVC: MVC stands for Model View Controller in J2EE is a software architectural pattern for implementing user interfaces.

ORM: ORM stands for Object relational mapping Object-relational mapping (ORM, O/RM, and O/R mapping) in computer science is a programming technique for converting data between incompatible type systems in object-oriented programming languages.

V/TI: Vehicle Track Inspection is nothing but an automated vehicle which rolls on the track and collects the data about the track and store in the database for further use.

Summary

This chapter gives a basic idea about the problem, objectives of the project, nature and significance of the project at the business, problems being solved by the project and the assumptions made throughout the development of this project. It also explains the technical abbreviations and other terminology used in this document.
Chapter II: Background and Review of the Literature

Introduction

This section gives more detailed information about the background of the problem and the literature study. This helps the user to get better information about the problem and the studies made to propose this particular solution.

Background Related to the Problem

Rail Track inspection is the practice of examining rail tracks for flaws that could lead to catastrophic failures. According to the United States Federal Railroad Administration Office of Safety Analysis, track defects are the second leading cause of accidents on railways in the United States. The leading cause of railway accidents is attributed to human error. The contribution of poor management decisions to rail accidents caused by infrequent or inadequate rail inspection is significant but not reported by the FRA, only the NTSB. Every year, millions of dollars are being spent to inspect the rails for internal and external flaws. Nondestructive testing (NDT) methods are used as a preventative measures against track failures and possible derailment (Office of Railroad Safety, 2015).

There are many effects that influence rail defects and rail failure. These effects include bending and shear stresses, wheel/rail contact stresses, thermal stresses, residual stresses and dynamic effects.

Defects due to contact stresses or rolling contact fatigue (RCF) are Tongue lipping, Head checking (gauge corner cracking), Squats—which start as small
surface breaking cracks. Other forms of surface and internal defects include Corrosion, Inclusions, Seams, Shelling, Transverse fissures, and Wheel burn.

One effect that can cause crack propagation is the presence of water and other liquids. When a fluid fills a small crack and a train passes over, the water becomes trapped in the void and can expand the crack tip. Also, the trapped fluid could freeze and expand or initiate the corrosion process. Parts of a rail where defects can be found are Head, Web, Foot, Switchblades, Welds, Bolt holes.

In the web application being developed all this data is given and then the application analyses this data according to the algorithm and informs the operators to perform the inspections.

RTCM enables field inspectors to record track inspections without the use of paper. The RTCM provides a means for track inspectors and maintenance personnel to manage inspection schedules, record track related defects and maintenance activity, and synchronize all field information with a centralized data management system. Additionally, the RTCM can quickly create regulatory and management reports.

The four most commonly occurred problems are:

1. Lateral track irregularities.
2. Short-wavelength vertical track irregularities.
3. Long-wavelength vertical track irregularities.
4. Wheel/Rail impact.
Literature Related to the Problem

Data from the tracks is collected from unattended vehicles known as V/TI (Vehicle Track Interaction) monitors. This data comes from the V/TI when the contact takes place between rail vehicle and the track. Compact sensors, GPS and other wireless satellite communication devices are used to map the contact between vehicle and the track and convert it into usable data. This approach is known as “train path free” approach. Hence it eliminates the need of specialized personnel to collect the data or special equipment to collect the data. So it is cost effective, time effective and also facilitates the 24 hour monitoring of the track and real time reporting of the information and safety conditions. The data being collected is stored in the database and the application that is being built will be using this data to generate reports, charts and other user friendly format for the detailed analysis. So this can be used in long term analysis and degradation monitoring which helps in maintenance planning. Measuring the track frequently and collecting the data for the continuous monitoring can be overloaded with data some times and they are optimized via wireless communication to the application and reporting capabilities are leveraged for the use of such results. Some of the capabilities include

- Linear and geo referencing of location detections to track assets.
- Automated distribution of notifications to MOW personnel.
- Simple to use, enterprise web-based reporting system with GIS and tabular based reporting tools for data analysis.
Java/J2EE is the programming language used to develop this web application. J2EE advanced technologies being used for this project are spring (Model View Controller) framework and Hibernate (ORM framework).

The Spring web MVC framework provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

- **Model** is the object which has the properties and encapsulates the data for the specific object which generally consists of.
- **View** is where all the data gets presented in whatever forms required (depends on the front end technology used). Basic output will be in the format of HTML where the client's browser can interpret.
- The **Controller** is responsible for processing user requests and building appropriate model and passes it to the view for rendering.

![image](image_url)

Figure 1: Spring MVC Framework Architecture
**Hibernate ORM** (Hibernate in short) is an object-relational mapping framework for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions.

![Hibernate ORM Framework](image)

Figure 2: Hibernate ORM Framework

In the front end **JQuery** is the frame work used to display all the data in the form of charts, tables, and user friendly format. JQuery is built on top of JavaScript and is very fast, small and feature rich. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.
Canvas JS plugin in the JQuery helps in building the charts and graphs from the raw data to a web page with very few lines of code. Twenty-four types of charts can be drawn using this plugin which include line, column, bar etc. Tens of thousands of data can be plotted without any lag.

**MySQL** server database is used to store all the data about rail track and its details. It is a RDBMS database which allows users to store the data in a very effective and efficient manner. Whether the data being entered manually or the data coming from the V/TI comes to this database. MySQL is relational database management system where the objects can be stored as is and can be retrieved as is. Java model objects can have direct representation in the database.

This can be integrated with Java on any platform like Windows, Linux, and Unix etc.

The following figure depicts the integration of Java, Hibernate and MySQL database interface.

![Figure 3: Spring Database Integration](image-url)
Literature Related to the Methodology

**Lean Development (LD) Methodology.** Lean Development methodology is employed for this project. As this project comes more under change tolerant software methodology suites this project perfectly. As this project involves dynamically changing software and up to date requirements application needs to be adjusted and modified. So this Lean Development focuses more on the dynamic stability which is similar Scrum embracing the controlled chaos. According to originator of this Lean Development, Bob Charette, the measurable goal of LD is to build software with one-third the human effort, one-third the development hours and one-third the investment as compared to what SEI (Software Engineering Institute) CMM Level 3 organization would achieve.

Twelve rules which form the Lean Development are:

1. Customer satisfaction is the highest priority.
2. Providing the best value for the money always.
3. Regular interactions with customer and participation influences the success.
4. Team work and effort is the key to LD.
5. Everything is changeable.
6. Domain, not point, solutions.
7. Complete, don't construct.
8. An 80 percent solution today instead of 100 percent solution tomorrow.
9. Minimalism is essential.
11. Product growth is feature growth, not size growth.


Figure 4: Lean Development Flow

Common principles Lean Development follows are:

1. **Eliminate Waste**: In the software development industry the most commonly identified wastages are unnecessary code or functionality, starting more than can be completed, delay in the software development process, unclear or constantly changing requirements, bureaucracy, slow or ineffective communication, partially done work defects and quality issues, task switching (Waters, 2010).

2. **Build Quality In**: One of the techniques used in this is **Pair Programming** which helps to avoid issues by combining the minds of two developers to each specified task. So collective, combined experience of two developers can be used to yield better productivity and better solutions. Second
technique is **Test Driven Development** which avoids most of the quality issues by writing tests before writing code. So basically it is nothing but Test QA writing the test cases before the code is developed (Waters, 2010).

3. *Create Knowledge:* In software development one developer that works on everything and does the work productively creates a knowledge that no one can beat.

4. *Defer Commitment:* Decide as late as possible, is the key for this principle. This particularly applies for decisions that are irreversible, or at least will be impractical to reverse

5. *Deliver Fast.*


7. *Optimize the whole.*

**Summary**

This section gives the detailed knowledge about the problem background information and literature information about the software being used to develop the application and also the literature information related to the methodology of the project. Detailed implementation of the project will be discussed in next chapters.
Chapter III: Methodology

Introduction

This chapter gives detailed knowledge about the project framework and design. Implementation of the project and actual development of it is also discussed here. In addition to these data collection and timeline of the project is also discussed here.

Design of Study

The main aim of the project is to develop an application which displays the rail track data in the user-friendly format (graphs, diagrams). So this project deals with new programming interfaces, backend data collection and also storage of the data. Data collected here will be done in two different ways. One way is worker working on the rail track directly enter the data and store it into database and the other one is V/TI running on the track collected the data from the track as it is making contact with the rails. So this application helps users to easily find the defects on the tracks, exact position and time on the track where the inspection to be performed without having to manually find the defects and note down the timings.

In order to get these results in more accurate and more efficient way entire process has been divided into three phases.

First phase is getting the data from the sources. As mentioned above data is collected through two different ways, one is by the rail worker entering the data into the application and storing it in the database and the second one is through the V/TI machine getting the data from the rail tracks and storing it into the database. All this
data needs to be validated before it is being entered into the database as the same data will be used by the application to display the reports, graphs and user readable data about the defects on the track and also the inspections to be performed on the tracks. For this batch jobs are created. Batch job is nothing but collecting all the data from the sources and running all the data at the night times with all the validations and store it into database. This process runs as a job once the batch creation is completed. Validations that are performed in this batch job before storing the data into database are as follows

- Is the data entered has all the required fields?
- Is the data format correct (date field should consist of date format)?
- Is the track data corresponding to the respective track?
- Is the time and date noted for the inspection?

After getting the data and storing it in the database, developed application uses that data to transform it into user readable format and displays it in the form of graphs, charts and bar diagrams. It also tells the user the condition of the track and the specific inspections to be performed at specific locations. Hence this data has to be very accurate that the inspections should be done in timely manner and at exact locations. If an inspection is already performed user should be notified of that to eliminate duplicate inspections.

So in order to achieve this a combination of quantitative and qualitative approach is followed throughout the process. Key objective in all the phases of the project is to provide the reports more reliably because the processed data is directly
reaching the customer. Data with errors can give frustration to the customers for additional or wrong charges that are billed to them. So throughout the process, the approach is a combination of both qualitative and quantitative approaches. The reason for this is as mentioned above, the company is the top most propane supplier as a result it got millions of customer records eventually millions of meters. So the data that needs to process is huge and results should give error free data to the customers.

**Approach followed:** Life cycle used to develop this process is **systems development life cycle (SDLC)**, also referred to as the **application development life-cycle**. This is a term used in software development systems and software engineering to describe a process which includes all the stages from planning, designing, developing, testing and deploying and application. SDLC concept can be applied to a wide range of software and hardware integrations and also to system consisting of both.

![SDLC Diagram](image.png)

Figure 5: SDLC
As mentioned above in Chapter II, methodology involves four different phases which has different purposes as follows.

a) **Project Assignment and Defining Roles.** This is the primary stage where scopes of the project and also the project limits are observed and described. Background of the project is thoroughly explored detailed description of the project is updated. In order to do this detailed DBRs (Detailed Business Requirements) are required and understood.

Different tools required for the project are identified at this stage. Then availability of those tools and also the cost effectiveness is identified. Then the intrigued individuals who are part of this project are identified and all the documented details are provided. So altogether end user should be made very clear about the requirements and business team should be clear about the expected output. The tools involved in completion of the project are mentioned in Table 1.

Table 1: List of Technologies Used

<table>
<thead>
<tr>
<th>Tool</th>
<th>Version</th>
<th>Source</th>
<th>Used For</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySql</td>
<td>8.0</td>
<td>mySQL</td>
<td>This database is used for data storage.</td>
</tr>
<tr>
<td>Spring MVC</td>
<td>3.1</td>
<td>Spring Framework</td>
<td>This framework is used for extracting transforming and loading data.</td>
</tr>
<tr>
<td>Java/J2EE</td>
<td>1.79</td>
<td>Java Opensource</td>
<td>This language is used for coding.</td>
</tr>
<tr>
<td>JQuery</td>
<td>4.1</td>
<td>HP</td>
<td>This is used in the frontend to display the data and defects.</td>
</tr>
</tbody>
</table>
Hence the key activities include defining the scope and establish scope dimensions. Agile methodology is being followed in this specific project where project assigns various roles such as team lead, business analysts, developers and testers to the team members. This competes the team formation and assigning project.

b) **Project planning and Sprints.** Initial project meeting was held to discuss various life cycles of the project with all the team members. This also included the design of layout of the project and design of the database to store all the data needed for this.

Appropriate and quality data is very much essential for this project as discussed in the data analysis stage. Hence validations for the data are planned well advanced to rectify the before it is stored in the database. Gaps in the data are also identified and is essential to achieve project bounded deadlines and its quality.

This project is done in two week sprints where a daily scrum meeting was conducted for all the team members every morning to discuss the tasks they have accomplished yesterday, obstacles they are facing today, and tomorrow’s goals. At the end of every scrum project is presented to the business to compare the requirements and output.

Work is break down into small parts and tasks are assigned to team members for every sprint. Work break down structure is designed by manager. Following is the work break down structure planned.
c) **Analyzing Requirements.** Once all the data is collected the project starts with designing a database which stores all this data. Designing a database includes creating tables and developing a relation among them. Once the database design is done functional lead, business analysts, project manager and development team will meet to check whether the customer specifications to develop the web application could be developed within estimated time and project manager assigns tasks to the development team. Various software, hardware and tools that will be required to develop the software will be identified; manage and track the progress of the web application development.

d) **Code Design.** Tasks of creating a scripts for developing database and then inserting the data has been assigned by each specific team leader to their teams. Team chooses the best design for the software development and it can used by anyone universally. Java has been employed as the primary Object Oriented Software to develop this application. Web
application layout prototype will be designed using HTML, XML and JavaScript. Each member of the team will be assigned specific functionality of the web application development and certain estimated time will be allotted to develop the functionality of the web application.

Microsoft Visio has been used to design the code. Below is the sample of the project designed using Visio connecting the flow of request in the code.

![Visio Diagram](image)

**Figure 7: Code Flow Development Using Visio**

e) **Code Development.** A code development practice will be chosen during the team meeting which has to be followed by each and every team member. In this code is developed by each team member individually for their specific assigned task. Developed code must be unit tested and pass
through the team leader check before it is committed. GitHub has been employed as the version control tool where the team members can commit their code and other team members can retrieve it.

f) **Code Testing.** Each developer performs unit testing before they check in their code or commit their code. When the code is deployed into test environment QA and testing team comes into picture where they perform aggressive testing. Each and every team member’s code must be checked before they are actually delivered. For User interface Selenium IDE testing and for code JUnit testing has been employed. Generally, development and testing is done parallel in order to eliminate delayed code errors, bugs and functionality mismatch.

   System testing has been performed which involves both the functional and non-functional testing.

   Functional testing: This is the initial testing where risks are analyzed based on the requirement specifications. Six steps involved in the Functional testing are:

   1. Expected output according to the business is identified.
   2. Based on the requirement specifications, input data creation is identified.
   3. Output determinations according to the functional specifications.
   4. Already prepare test case execution.
   5. The comparison of actual and expected outputs
6. Customer needs are verified as per specifications.

So finally with the functional testing what the developed application does is verified and answered.

Non-Functional testing: This is the second type of testing where the developed software application is tested for its non-functional requirements. These include how the application operates rather than how the application behaves. In functional requirements application is tested to check whether it meets the specific functions described or specified by the user. This non-functional testing is exact opposite to this. So this may include reliability, scalability, load testing, performance testing, stress analysis, portability testing and even more. So with this non-functional testing the main answer going to be answered is how the application works.

Code is developed and deployed to test environment where the testing takes place. So there is a scope of having different set of data or different environment properties between the Test and Developer environments. So both the environments should be as much similar as possible. Test team is responsible for testing and writing the test scripts. Test team log the defects as they find them and assign them back to developers for fixing them. Below is the image showing the defect being logged
g) **Improve Phase.** After the QA team finds the defects and log them, they are assigned back to the developers responsible for that. At this stage initially it is assessed whether the defect is due to the code or test error or environment issue. Once it is determined development team works on fixing the bug and reassign it back to the QA team for retesting.

After the Test environment there are two other environments this application is tested. They are Pre production and Production stages. This gives ample scope to improve and get the error free application.

So to have the right decision made, testing team may employ small testing cycles called “PDCA” or Plan Do Check Act. This helps in redefining the ideas of both the developers and QA team and also collecting the feedback from the business. In order to find out whether these small
improvements are effective and reliable these small cycles are the great way.

In order to have this improvements getting successfully implemented careful planning is required. So here both the developer and QA teams consider having the frequent communications, training and also planning. The more time the team spends on planning the faster they reach total adaptation to the improvements by their process

h) **Deployment and Maintenance**. As per the customer requirement, once the basic functionalities of the web application are developed, and all the defects raised by the QA team are fixed and the web application software is deployed and the web application goes live so that it can be the actually used by the end users of the client. If there are any errors or bugs while using the web application, these are reported to developers/testers. The assigned developers and testers take the responsibility to solve the issues. Once high priority issues are solved, software releases are made to effect the changes made to the web application.

**Data Collection**

As discussed above data has been collected in two different ways. One is worker working on the track enters the data directly into the application about the track and its properties. Second one is through the V/TI vehicle which runs on the track and continuously collects the data from the track and stores it into database.
Hence these can be categorized in two different methods of data collection called Quantitative and Qualitative Data collection methods.

First method of collecting the data, which is entered by the worker manually from the track comes under Quantitative data collection method. This is because worker enters the data randomly from any location and about any information of the track measured using any random instruments that fit diverse experiences into predetermined response categories. Results produced from this data collected can be easily summarized, compare, and generalize.

Here workers are randomly assigned to different tracks to be tested. So the data collected depends on location, situation characteristics and also the instruments used for measurement. Hence the results also depend on the size of population used to collect the data.

Typical quantitative data gathering strategies used to collect the data:

- Trails performed on the tracks.
- Observation of the behavior of tracks at most traffic locations (e.g., stress analysis at the location where there are at least 10 passages at the specific location of the track).
- Using validation techniques to retain only the relevant data.
- Comparing the results with previously determined answers.
- Surveying the same track at same location using different techniques.
Second method used for collecting the data is Qualitative method. This method played an important role in collecting the useful and error free data. As V/TI has been used in this method to collect the data validations are done at the same time data is being collected and then immediately stored in the database. Hence this data played key role in understanding the process behind observed results and their way of collecting data. Hence this method of collecting the data is used to improve the quality of collecting the data from the above method. This helped in designing more complicated validations that can be used in collecting the data using quantitative approach. There are different methods in collecting the data using this qualitative approach. They are:

- As the V/TI is collecting the data they tend to have automated validations and have less structured protocols (i.e., users can modify the way it is collecting the data or iterate through validations to obtain the relevant data).

- More number of iterations can be performed as this is fully automated process. Same track and at the same exact location can be surveyed many number of times to obtain the most accurate data. This helps in resolving particular issue and also to clarify concepts or check the reliability of data.

- Multiple data collection methods has been used to increase the quality of the data and also its reliability. This method is known as triangulation where data is collected through various means to check the authenticity of their results.
So all the data collected is not specific to any single track and these V/TI’s run all the night on different to collect the data.

**Data Analysis**

All the collected data after the validations are done are stores in a MySQL database. Application which is developed analyses the data using different techniques according to the business requirements and project the data in a user readable format. Data entered into the tables represent a model object and its properties. So Hibernate is the tool used to retrieve the data in the form of an object and display it according to the requirements. To analyze the gathered data, data analysis techniques were performed.

Data mining technique has been used to model the data rather than describing the data. BI (Business Intelligence) team focused more on the predicting the data outcome rather than purely describing the data. Two types of analysis are performed EDA (Explorative data analysis) and CDA (Confirmatory data analysis). Main difference between both of these techniques is EDA concentrates more on discovering the new features in the data and CDA concentrated more on the analyzing and confirming the existing data hypotheses.

Data collected from the both the sources as mentioned above goes into the same place before going through different validation checks. Data cached here will be process through a nightly batch jobs that run throughout the nights and puts them into the database in the required format. Both these techniques were then used to
retrieve the data to transform it into a simply understandable form, habitually with visual graphs, without using any statistical model or a formulated hypothesis.

**Budget**

All the costs that were incurred during this project like licensing, server costs, equipment costs, database installation and handling costs, resources, maintenance and other miscellaneous costs were covered by the company.

**Timeline**

Time line for the complete implementation of this project (Apr 2015) was about 10 months (proposed on OCT 2014). Total implementation was divided into seven different phases.

- Project Idea Formulation and Research……………………September, October 2014
- Project Proposal Write-up.................................................October 2014
- Gathering the requirements..............................................November 2014
- Analyzing the requirements..............................................November 2014
- Designing the code.........................................................December 2014
- Developing the code......................................................December 2014
- Testing the developed code............................................January 2015
- Deploying the tested code...............................................January 2015
- Evaluating the Web Application........................................April 2015
- Project Write-up...........................................................May 2015
- Final Defense...................................................................October 2015
Total duration of this project is 12 months and went live on May 2015 with zero percent defects.

Summary

This section discusses about the study made prior to the start of this project and the methodology followed in implementation of this project. It also discusses about the collection of the data on which this application is going to run and methods, tools used to analyze the data collected.
Chapter IV: Data Presentation and Analysis

Introduction

This chapter explains about the data presentation and analysis of the data that is being presented. Data presentation is a key aspect of this project. It is very much important to arrange the data in meaningful manner to portray the analysis of the presented data. With the help of the presented data users will be notified of their scheduled inspections and locations on the track. In this chapter it is clearly explained that how the data is presented and what way the data has been analyzed.

Data Presentation

As discussed in the above chapter data has been collected in two ways. First one is through the manual process. In the developed application there is a page with all the required fields to be entered and also validations done automatically on the front end side of the application. Below is the image showing the page where details of the track are entered and validated.

In one part of the screen fields are present to enter the inspections details performed on the tracks and on the second part of the screen there are fields present to enter the defects details for that track. For each inspection there is a possibility of entering the defect related to the inspection. After entering the details there is an option for saving these fields into the database.
Figure 9: Sheet for Entering the Data

Once the data is collected this is displayed according to the inspection or defect or schedule selected. Once any one of these is selected details about that will be displayed in the tabular format. This table includes division or sub division where the defect or inspection occurred, scheduled timeline, asset number and location of the detail in terms of mileposts. Below is the image displaying the results of the scheduled inspection with the tabular details? In the last column is the calendar code where it displays the codes for the scheduled inspections. For example if it says W1_3 inspection has to be performed thrice a week. And if it says M1_20 for every twenty days once or first week of the month whichever comes earlier.
Figure 10: Data Display in Application

Data Analysis

In this stage, Data collected using various means by using different techniques and by using different methods are to be divided into small pieces of data where the application can analyze and make an inference. As seen above application uses the collected data to report the results to the user and have achieved an unmistakable part to accomplish evaluated results. Following Business and user requirements are the key things responsible to achieve these results. Below is the chart showing the defects found from the track after the application has analyzed the data and scheduled inspections. This is in the time frame of April through June. Red bar indicates the total cumulative defects activated and green bar indicates total
cumulative defects terminated. Black bar indicates the total number of cumulative defects still active. This is high in this time frame as the web application was in development and testing stages. Hence total number of defects have been increased in this time frame.

![Change in total defects: +1](image)

**Figure 11: Application Data Defects**

After this time frame when the application was moved to production stage after all the defects in the code were closed there has been a lot of improvement. Below is the chart showing defects that the application has been produced and the kill rate after each defect in the code is fixed. This is in the time frame of month July through August.

Red bars indicate the defects arrival rate and green bars indicate the kill rate of those defects. At the end of month August kill rate has been too high and literally
there are no new defects raised. These are defects from the results produced by the application after analyzing the data and producing the output.

![Defect Arrival and Kill Rate Chart](image)

**Figure 12: Defect Arrival and Kill Rate**

Below is the track of defects in the code as it is being developed. Primary aim of the project is go error free in to the production. As the figure explains defects in the development stage, testing stage, new defects raised and closed defects. At the end of the process where application goes into production it was an error free code.
Figure 13: Defects in Code

**Report analysis generated by the application:** Following diagram shows the total number of errors occurred on different month’s data during the testing process. These are the errors occurred from the reports generated by the application monthly wise. April is the month where the code has been to UAT testing where all the defects has been rectified as they are raised and generated new reports.
Figure 14: Total Number of Errors from the Application Produced Reports

Track analysis month wise: Three tracks have been identified and reports have been generated on these using the application. Below is the chart explaining number of errors occurred from the month of December through April. Defects are identified on a regular basis and are very consistent and proved to be very effective.
Summary

This chapter explains the way how data flows in the business and how this project can be used to analyze the company’s data. This automated reports which runs as jobs can generate these results at given point of time. Next chapter discussed about the results achieved by this project at the client location.
Chapter V: Results, Conclusions, Inferences, and Recommendations

Introduction

This chapter discusses the results obtained from the data collected and effectiveness of these observations for the organization by implementing this project. This chapter also discusses the recommendations and conclusion of the project and how this project answered questions raised at the initial stages. Potential future works and recommendations for the extension are also discussed.

Results

At the end of this project, robust web application has been developed which handles the huge chunks of data and utilizes the data to transform it into user friendly or readable format more quickly and efficiently. This application is designed in such a way that it can be operated from a system or the special gadgets that are provided to workers by the organization. This is because workers need to enter the track related data from the working sites where they can easily carry gadgets along with them. So this application acts as an interface for various platforms and yet stores the entered data into the specified database. Again by using this data this application transforms it into graphs, charts and diagrams and notifies users about the tracks and their properties. It automatically notifies users where to perform the inspections and when to perform. So this application helped the organization in eliminating various legacy systems which are being used to perform the tasks independently. This application brought all the applications under the same roof. Reports generated by this
application can be analyzed and users can constantly keep track of tracks and improve the services.

**Project Questions**

1. *What are the ways to identify the defects in tracks?*

   Schedule of maintenance for the tracks is maintained throughout the year. Data collected through the ways discussed above (Manual and V/TI) consists of the rail track information such as the last inspection performed on the track, next inspection to be performed, more stress area and so on. With the help of this data web application developed creates a schedule and notifies the workers to perform the checks on specific tracks at specific locations at specific time periods.

   So defects in the tracks can be easily identified with the help of these regular and periodical inspections. As dashboards generated by the application are very dynamic they change according to the data entered. Hence this application provide the ability to process the huge data and convert them to instructions in the form of charts or graphs which helps in timely inspections on the track at exact specific locations.

2. *Where to perform schedule of inspections for specific tracks at specific location?*

   Initially certain specific locations are identified where there is a more stress and more traffics on the tracks. At these specific locations checkpoints are formed and ensure that data is regularly collected from these locations.
Hence data will be regularly collected and stored in the database using both the data collection techniques. Application uses this data to analyze the track condition and provides user with a specific time period to perform the inspections.

3. Was RTCM providing correct information about the data and its assets, if yes were there any failures in identifying defects in tracks?

RTCM application provides user the collected data regarding the tracks in the explanatory manner and helps in scheduling the inspections at checkpoints at regular time intervals.

   Considering the inspection performed on the track at the specific location and at the specific time RTCM schedules the next inspection for that same location depending on the stress level that the location is being exposed to. Various inspections have been scheduled and performed and there was only 1% of errors in total. This 1% error is schedule being mis-timed and not performed where it is supposed to be. This is because stress level was very minimal at the starting stages and it is increased to high levels at the later stages of project.

   Hence as a result of this stress levels are also being analyses from time to time which resulted in perfect inspections and achieved 0% error rate.

4. Was RTCM really helpful in providing data on field where there is no internet and helping in getting the relevant and valid data?
As discussed in the data collection stage there are two ways of collecting the data from the track. One is though manual process where workers have to go on to the track and input the data into system. Second one is through V/TI vehicles which run on the track and intake the data. So coming to the first part where workers have to collect the data, there need to be a perfect validations as the data entered is going to be used by the application for the analysis purposes.

So in the developed application where the workers enter the data validation check points are coded which tells the worker exact format of the data to be entered, required fields and the exact locations. So the data entered here will be loaded into the cache and nighty batch jobs run to insert this data into the database. Hence RTCM application helped workers to input the data without them having them to verify and check the data.

5. *Does RTCM really reduced the labor required to access information commonly used to analyze track conditions?*

Before RTCM there are various legacy systems working for each independent task. For example to keep track of the inspections schedules there was one legacy system, to keep track of defects one and collect the data one. So with the help of this RTCM application need for these legacy systems was completely eliminated which in turn reduced the amount of labor required.
Right from the data collection, analyzing the data, keeping track of the inspections schedules this single application can handle and perform much better than legacy systems. This single application can also be accessed via tablets given to workers and work in offline mode as well.

**Conclusion**

There are no tools existing in the present market which specially used to monitor the rail tracks and their defects. So there is always a need of specific applications which can help organizations keeping track of the rail tracks and their properties.

Hence RTCM was the solution proposed to handle all these requirements. This combines the software tools with the mechanical conditions to calculate and schedule the inspections on the tracks. In order to achieve this robust coding language is required which can be run on any platform. Hence Java language is employed for this which can be written in one language and run on any operating systems. All the logics, calculations and conditions are coded in this language which gives the desired input based on the input given. So the data is processed in this language. Next step is to present the processed data. So for this front end tools such as JQuery is selected to display the data in the required or desired format. MySQL database is used to store the vast amount of data from tracks.

With the help of all these technologies RTCM application is developed which is dedicated to the organization to stay competitive and updated with the rail industry. With this company was able to regularize the process of inspections and also to
reduce the extra labor. With the help of this insight within the industry was increased with which they can increase their profits, identify and decrease potential and ongoing losses and comply with the industry regulation and standards. This tool many capabilities, can handle vast amount of data from different sources. As all the technologies used are common one there is always a scope of expanding its capabilities in the future.

**Recommendations**

This project mainly focuses on developing reports and scheduling inspections using the application developed with the Java programming language and MySQL as background database. Here are few recommendations to implement similar kind of project.

- Data is collected from different means and data stored in the database is always validated before it is stored.
- Users should get hold of the application and its developments at the end of every sprint and provide their inputs if there are any changes.
- Data collected will be stored in the database after it is validated. It is done by nightly batch jobs which should run every night to keep up the updated data.
- All the environments where the application is running should have similar set of data and configurations. Any changes to one environment should be notified and changes should be done in other two.
Scheduled inspections should always be performed and should be entered into the system once the inspection is done to schedule next one.

Most calculations done are in the java code. So they should be done at the back end to promote faster responses.

Any questions with the application user can go to help provided in the applications. Server related issues can be directly logged into the system.

Java code is more reliable and less cost effective, as it is an open source and can be run on any platform which mean run time may be lesser than the custom code because the standard code gone through various levels of performance tuning.
References


