Software Requirements Specification (SRS)
Paint Defect Analysis Tool (PDAT)

Authors: Team 12 -- Kyutaek Park, Christopher Cho, Kevin Ahn, Jon Hayward, Vikram Thakur
Customer: General Motors
Instructor: James Daly

1 Introduction
In this document, the Paint Defect Analysis Tool, or PDAT, shall be covered in complete detail. The reasons for its production, the parts one would need to use it, the requirements it must meet, and detailed specifications will all be discussed by this document. Also, the features of a prototype will also be covered. This document will describe the entire PDAT system and its major functions as well as the development process behind it. It will introduce potential constraints and discuss the dependencies of the system regarding the defect analysis occurring in an automotive paint line. Technical models are also provided and a prototype is discussed in order to create guidelines as to how the system will be built.

1.1 Purpose
The purpose of this document is to catalog the user requirements, tools required, and motivations for the paint defect analysis tool being developed by the team. This document is meant to help facilitate and reach an agreement between the end user and developer, the system architect and the implementer, as well as the programmer using the module and programmer implementing the module.

1.2 Scope
The software being produced is a system that is meant to automate the aggregation and generation of an automotive paint defect analysis tool. This does not involve replacing an employee on the plant floor checkpoint areas, however it does eliminate the use of a paper diagrams to record paint defects. PDAT is the application tool being built and the domain of its development is only program being developed. The new implemented system requires an employee to use a tablet to input any defects that were previously recorded by hand. These reports can be automatically collated to create quality analysis reviews and reports. By creating a digital system, we enable an increase in speed, reliability, security, and portability. The objective of the system is to improve overall efficiency in the process of creating analysis reports. The software will not be automatically gathering input data via sensors or other aids, it will still require an employee to monitor vehicles. The software will also generate reports by request of the user and be able to apply each certain filters to organize each report. Ultimately, the elimination of a reliance on paper and manual analysis will result in a streamlined tool and service.

1.3 Definitions, acronyms, and abbreviations

<table>
<thead>
<tr>
<th>Required Terms, Definitions, or Abbreviations</th>
<th>Definitions</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit</td>
<td>The electronic equivalent of the paper diagrams that a client analyst will fill out when inspecting vehicles</td>
</tr>
<tr>
<td>Defect</td>
<td>Paint defect, flaws, or imperfections in the paint and finish</td>
</tr>
<tr>
<td>E-Coat (Electrocoating)</td>
<td>Station where inspections are performed. Process used for coating vehicle</td>
</tr>
<tr>
<td>Employee Verification</td>
<td>A process to authenticate users. This will determine how much access the user is given based on their status</td>
</tr>
<tr>
<td>Horizontal vs Vertical (defect locations)</td>
<td>The left and right sides of the vehicle are vertical, while the roof and hood are horizontal</td>
</tr>
<tr>
<td>Paint Defect Location Abbreviations/Descriptions</td>
<td>Specify abbreviations used to describe the location of the paint defect</td>
</tr>
<tr>
<td>PDAT</td>
<td>Application name, acronym for Paint Defect Analysis Tool</td>
</tr>
<tr>
<td>DPU</td>
<td>Defects per unit</td>
</tr>
<tr>
<td>Polish Deck, E-Coat, Prime Review Checkpoints</td>
<td>Different checkpoints to analyze the quantity and severity of the paint defects</td>
</tr>
<tr>
<td>Report</td>
<td>Once a sufficient sampling of vehicles is done, the analyst then collates the collected data and produces a daily report on the nature of the flaws found</td>
</tr>
<tr>
<td>Unit</td>
<td>Single vehicle, paint defects are recorded per vehicle</td>
</tr>
<tr>
<td>User Interface (UI)</td>
<td>The means by which the user and a computer system interact, in particular the use of input devices and software</td>
</tr>
</tbody>
</table>

### 1.4 Organization

The following portions of the SRS document will contain an overall description of the Paint Defect Analysis Tool, models that describe the software in a more technical depth, a functionality specification for the prototype and the references used throughout the development of the SRS document.

2 Overall Description
   2.1 Product Perspective
   2.2 Product Functions
   2.3 User Characteristics
   2.4 Constraints
   2.5 Assumptions and Dependencies
   2.6 Appropriation of Requirements

3 Specific Requirements
2 Overall Description

In this section, the aspects of PDAT not directly tied to how it performs will be discussed. The context of the product, PDAT’s functions, the users of PDAT, the team’s assumptions, constraints for PDAT, and requirements that will not be fulfilled by PDAT are discussed at length in this portion of the document.

2.1 Product Perspective

The PDAT will provide a way to improve the data entry method of paint defects on an automotive line. This will contribute to the overall paint assembly line system and will utilize other systems throughout the manufacturing plant. The solution that the analysis tool will provide must have specific characteristics. A friendly user interface is essential to increasing efficiency of data input. This also allows for a wider range of skilled employees to interact with the software.

The user interface will be directed towards the defect recorder. Thus, the application will have an authentication page, a page selecting the type of vehicle, and a page that will allow the user to submit defects on a specific vehicle. On each review line, a client analyst will log into the application, navigate the menu and select if they would like to view a report or log data entries. If the analyst decides to log data entries then they will proceed to select the current vehicle they are analyzing and begin to mark defects. Each defect type will be chosen via a toolbar and it will be recorded upon press to the vehicle diagram on the screen. This data will be automatically saved, so the employee may feel free to move between different vehicle diagrams as they come down the line. After the data is saved, it will contribute to the different reports that are then generated automatically. These reports will also be able to be accessed whenever needed via the opening application menu.

By automating the paint defect analysis tool, the program will eliminate unexpected difficulties such as unreadable handwriting or ambiguous defect marks. The tool must also meet specific performance requirements for both the web application and the report generator to create a seamless experience. The collation of the vehicle diagrams will be automated. The quality analysis reports must also be generated autonomously. These reports must be able to be created with any given time frame. To ensure the integrity of the reports, only users with correct security credentials will have access.

The PDAT tool will require synergy between multiple systems. To eliminate the use of paper, a web application will be used via tablet system. This ensures a more secure method of reporting and storing data. The analysis tool requires a connection to the plant’s network to interact with outside systems and improve communication. To further secure the reports, an employee credential system will need to be accessed in order to create reports and analyses. The report's data will be stored in a database system. A diagram of the system interactions with each other is shown below in figure 2-1.
Because the system relies on other systems and sensitive manufacturing data, there are constraints that exist. Secure access to data and the systems must be ensured in order to preserve integrity. Another form of constraints that exists is team constraints which include organization of time relative to other classes, meetings, and work load throughout the development process. Meetings and equal workload should be organized weekly and effectively between project development members.

**PDAT Data Flow Diagram**

![PDAT Data Flow Diagram](image)

*Figure 2-1 PDAT System Flow Diagram*

### 2.2 Product Functions

The major functions that PDAT will perform include data retrieval. It will store user inputs and also be able to retrieve access information for each user. Furthermore, data mining will take all the inputs, entered by the user using a tablet, and will output the data to be analyzed. Once the new data have been created, the software will generate charts to visualize how these values have been changing weekly. As per authentication, if the user does not have access to the software, the software will prevent the user from entering any data. This data will include location, type, and severity of defects. As per authentication, if the user does not have access to PDAT, it will prevent the user from entering any data. The customer
specified that PDAT would be compatible in all three of the GM plants, so the output of the data must account for different systems being used to generate the desired reports in all three plants. Also, the functionality specified by the user include a friendly user interface and collation of the vehicle diagrams.

2.3 User Characteristics

The system will be built regarding mainly two types of different users. The first user is the inspector that will be located on the line at a checkpoint. This user will look at vehicles and use the system to report defects in the paint. This person will be skilled with their eyes to recognize defects, and be able to use a tablet but is not expected to be extremely proficient with computers or have overarching analysis skills.

The other expected user is the analyst. This user will take the data gathered by the inspector, which will usually be the analyst, and then generate reports. The analyst will then comb through the reports to determine trends and find solutions for what is causing the defects. This person will most likely have a technical degree and will have extensive experience with computers.

2.4 Constraints

Due to PDAT’s reliance on external systems, an immediate constraint will be the reliability and security of those systems. For example, the login system will require that credentials will have integrity and correct security correlations. Also in order for the PDAT to update databases, network reliability is required. PDAT may be constrained due to safety concerns as well. A user creates and fills out an audit through an app on a tablet. The tablet must be able to run the app to provide a smooth and stable user experience. PDAT does not store data on the tablet directly but in a database elsewhere. To access this database, the tablet must have an internet connection to submit audits, request to generate reports, and view completed reports. For security concerns, only users with verified credentials have access to creating audits and dealing with reports. This authentication is verified by General Motor’s pre-existing employee verification system.

2.5 Assumptions and Dependencies

The hardware that PDAT will run on will primarily be a tablet that the client inspector will use when checking cars for paint defects. The system will also depend on the database that holds all stored reports. PDAT will also expect some sort of authentication system provided by each plant. PDAT will have to interface with file creation and a printer. User interactions with PDAT should be restrained to analysts making inputs of defects and then generating reports based on these defects.

2.6 Appropriation of Requirements

The current system is meant to interface a system in order for data entry by employees for defects and generate reports. In addition, PDAT will still require an employee to monitor vehicles to identify and enter the three data required for analysis: location, type, and severity. The employee must be verified through an existing authentication system in order to have the correct permissions to view data and manipulate data.
3 Specific Requirements

1. Log In
   1.1 User enters credentials in order to access analysis tool
   1.2 Enables a level of security so only authorized users can access the tool

2. Receive Paint Defect Data
   2.1 Data must include location of defect
   2.2 Data must include severity of defect
   2.3 Data must include the model of the car the defect occurred on
   2.4 Data must include the line number and plant the car was in
   2.5 Data must be able to be removed before it is stored

3. Store Data
   3.1 Data must be stored in a database
      3.1.1 The database must be secure
      3.1.2 The database must keep records indefinitely
   3.2 Data must be retrievable for the generation of reports

4. Generate Reports
   4.1 Report duration supported: daily, weekly, monthly
   4.2 Report types supported: Q.A, Audit, Summary Report
   4.3 Reports will contain collated data of defects defined by a user selected time frame

5. Log Out
   5.1 The system will log out user and send to login screen
   5.2 Enables a level to security to protect data integrity

4 Modeling Requirements

In order to understand program functionality and the generating of various types of reports a UML is shown in Figure 4-1a. The figure is then followed by an English description in Figure 4-1b which is the data dictionary in relation to Figure 4-1b. Figure 4-3 is a use-case diagram that describes the behavior of the system at a high level, followed by an explanation of each use case.
Figure 4-1a UML PDAT Class Diagram

Figure 4-1b PDAT State Diagram

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit</td>
<td>Summary of defects on single vehicle</td>
</tr>
<tr>
<td>Attributes</td>
<td></td>
</tr>
<tr>
<td>plantLocation : enum</td>
<td>Specific GM plant of vehicle being analyzed</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>checkpoint : enum</td>
<td>Checkpoint audit was performed at for defect</td>
</tr>
<tr>
<td>startTime : time</td>
<td>Start time and date recorded for audit</td>
</tr>
<tr>
<td>endTime : time</td>
<td>End time and date recorded for audit</td>
</tr>
<tr>
<td>numUnits : int</td>
<td>Total number of cars audited</td>
</tr>
<tr>
<td>DPU : int</td>
<td>Defects per unit of cars</td>
</tr>
<tr>
<td>Relationships</td>
<td>Audits can be accessed through the PDAT System. Audits also have information pertaining to Vehicle.</td>
</tr>
<tr>
<td>UML Extensions</td>
<td>QA Reports require audit information. Composition to Defect and Vehicle. Composition from QAResport.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defect</td>
<td>Current defect being recorded</td>
</tr>
<tr>
<td>Attributes</td>
<td></td>
</tr>
<tr>
<td>location : location</td>
<td>Location on diagram of present defect</td>
</tr>
<tr>
<td>type : enum</td>
<td>Type of present defect (i.e. adhesive, runout)</td>
</tr>
<tr>
<td>severity : enum</td>
<td>Severity of defect, from 1 to 5</td>
</tr>
<tr>
<td>Relationships</td>
<td>Defects are recorded on audits and eventually placed into a QA report</td>
</tr>
<tr>
<td>UML Extensions</td>
<td>Composition relationship to audit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DefectAnalysisReport</td>
<td>A defect analysis report</td>
</tr>
<tr>
<td>Relationships</td>
<td>A defect analysis report may be accessed by QAResports, and a PeriodChart may access a DefectAnalysisReport. Defect analysis reports also have information pertaining to Vehicle</td>
</tr>
<tr>
<td>UML Extensions</td>
<td>A defect analysis report is of type report and therefore derived from Report. Composition to QAResport and Vehicle. Composition from PeriodChart.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDAT System</td>
<td>Paint Defect Analysis Tool, application being used to generate reports and input defect data</td>
</tr>
<tr>
<td>Relationships</td>
<td>PDAT System creates and has access to audits, vehicles, and reports</td>
</tr>
<tr>
<td>Element Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PeriodChart</td>
<td>Duration based reports (1 week, 1 month etc), user selected time frame allows for aid in data analysis</td>
</tr>
</tbody>
</table>

**Relationships**
Audits can be accessed through the PDAT System

**UML Extensions**
A PeriodChart is a type of report and thus derived from Report. Composition to DefectAnalysisReport

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QAResult</td>
<td>Quality Analysis report to be generated</td>
</tr>
</tbody>
</table>

**Relationships**
Data from audit may be used in a QAResult. QAResults may also have information useful to a DefectsAnalysisReport.

**UML Extensions**
QAResults are a type of report and therefore derived from Report. Composition to Audit. Composition from DefectsAnalysisReport

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>Report to be generated</td>
</tr>
</tbody>
</table>

**Attributes**

- **startTime : time**
  - Beginning of time frame that report covers

- **endTime : time**
  - End of time frame that report covers

- **checkpoint : enum**
  - Checkpoint audit was performed at for defect

**Relationships**
Audits can be accessed through the PDAT System. Reports also have vehicle information. The overall system also may request any report.

**UML Extensions**
Composition to Vehicle. PeriodChart, DefectAnalysisReport, and QAResult are derived types of reports.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReportRequest</td>
<td>Request to generate report</td>
</tr>
</tbody>
</table>

**Attributes**

- **reportType : enum**
  - Beginning of time frame that report covers

**Relationships**
A report request generates a report and is requested by a user

**UML Extensions**
Association with User and Report

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Type of vehicle currently being analyzed</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>make : enum</td>
<td>Manufacturer of vehicle</td>
</tr>
<tr>
<td>model : enum</td>
<td>Model of vehicle</td>
</tr>
<tr>
<td>Relationships</td>
<td>Vehicle information is accessed by the PDAT system, an audit and a defect analysis report</td>
</tr>
<tr>
<td>UML Extensions</td>
<td>Composition from Defects/AnalysisReport and Audit, association with PDAT System</td>
</tr>
</tbody>
</table>

Figure 4-2b Data Dictionary (Pertaining to UML) [Organized Alphabetically]

The following describes the use cases of the system. The use cases describe the specific event(s) that may occur depending on what the user wants to perform.

![Figure 4-3 General PDAT Use Cases](image)

**Use Case:** Log In  
**Actors:** Inspector or Analyst (user)  
**Type:** Primary and essential  
**Description:** The user will start the system on their machine and then be prompted to enter their credentials. After they have submitted their login information, the system will authenticate the information and pass the correct users to the menu screen, and it will reject users that provide incorrect login information. Rejected users will not be able to enter any data to the system.  
**Cross Ref.:** 4  
**Use-Cases:** Authenticate User

**Use Case:** Log Out  
**Actors:** Inspector or Analyst (user)  
**Type:** Primary and essential
Description: After the user has submitted their login information, and the system has authenticated the information, they have access to the main menu where they can gather and input data, create audit, and generate reports. At the menu screen, they have the option to securely log out from the system which will prompt the user to enter their credentials.

Cross Ref.: 5
Use-Cases: None

Use Case: Authenticate User
Actors: System (Authenticator)
Type: Secondary and essential
Description: The user will start the system on their machine and then be prompted to enter their credentials. After they have submitted their login information, the system will authenticate the information and pass the correct users to the menu screen, and it will reject users that provide incorrect login information.

Cross Ref.: 4
Use-Cases: None

Use Case: Input Data
Actors: Inspector or Analyst (user)
Type: Primary and essential
Description: From the menu screen, they have the option to securely log out from the system which will prompt the user to enter their credentials. Once the user has chosen to input data, the system will have the user choose the vehicle to report the defect. After the user has chosen the vehicle, the user will be prompted to enter three data: location, type, and severity of defects.

Cross Ref.: 1
Use-Cases: Create Audit

Use Case: Create Audit
Actors: System (Creator)
Type: Secondary and essential
Description: With the data that the user has entered, the defects will be recorded into the vehicle, and there will be an option for the user to create an audit. This can be compiled later to generate a report.

Cross Ref.: 1, 3
Use-Cases: None

Use Case: Generate Report
Actors: Analyst (user)
Type: Primary and essential
Description: The user will request a report with a time period and report type. The system will gather the data, and then print the file it has produced.

Cross Ref.: 2
Use-Cases: Gather Data
Use Case: **Gather Data**

**Actors:** System (user)

**Type:** Secondary and essential

**Description:** The system will take the parameters for time period and report type and collect the required data from the database. Also, the system will composite audits from the selected time frame and generate a report. It will produce a file that will contain the relevant information in a presentable format so that the analyst will be able to determine trends.

**Cross Ref.:** 2

**Use-Cases:** None

The following two diagrams will be sequence diagrams that capture the two basic scenarios that the system will engage with. One scenario involves an inspector, while the other involves an analyst.

This sequence begins when the inspector logs on to the system. After being authenticated, the inspector begins to input data. He or she will create an Audit that collects defect data. The inspector enters the data, which is stored as Defects. The inspector finishes by finalizing the Audit, thus having the data store itself in the database. He or she finally logs out of the system, finishing the interaction with PDAT.
This sequence begins with the analyst trying to log in to PDAT. The system authenticates the user's credentials before granting access to the PDAT. Then, the analyst requests a report to be generated and gives the requirements for the report. The system then engages with the database and gathers and compiles the relevant data into a report before returning the report to the analyst. He or she will then log out of the system.

![Sequence Diagram for the Analyst](image)

**Figure 4-5 Sequence Diagram for the Analyst**

## 5 Prototype

The prototype includes a web application for inspectors to record their observations into a database, as well as a web application for supervisors to view and edit records inside of the database. The user interface contains buttons and menu options for displaying information and details about the inspection. The main device for the inspectors to operate on is a mobile tablet that they can carry around while they make their observations. In the sample scenario (Section 5-2), further details about the user interface and experience are detailed.
5.1 How to Run Prototype

In order to operate the prototype, a mobile tablet with a web browser is needed. The web application is functional on any operating system and does not require any additional plugins. A Wi-Fi internet connection in the facility where the application is used is also required. Lastly, a MySQL database is also needed to hold all records.

The prototype can be found at cse.msue.edu/~ahnkevin/prototype.html.

5.2 Sample Scenarios

A sample use case for the prototype can be from a GM automotive plant. Using a mobile tablet to access the application, an inspector logs in with their credentials (Figure 5-2a) to access the Action Selection screen (Figure 5-2b).

![Figure 5-2a Web Application Login Screen](image-url)
The user decides whether they want to either record defect data, generate a report, or audit individual defects. The user in this case selects the “Individual Defect Audit” button and is taken to a page to record defects (Figure 5-2c). On this page, the user can select the location of the defect on the car image, what type the defect is, how severe the defect is, and the number of defects in that location. Once the user submits all of their defects, they can exit to the main menu screen and continue the inspection process.
Figure 5-2c Individual Defect Audit

The user, with all their defects recorded, can now select to submit the data into the database by clicking the “Record Defect Data” button. This takes the user to the Plant Specification page (Figure 5-2d) to choose which plant, checkpoint, and vehicle model the user has inspected.

Figure 5-2d Plant Specification Page

Once all the fields are the filled, the report is generated and the user has completed their inspection process. In addition to recording their inspections, the user also has the option to generate a report. By clicking the “Generate a Report” menu option, the user is taken to a “Report Generation” page (Figure 5-2e) where they can then choose which report they want to generate.

Figure 5-2e Report Generation Page
Clicking any of the report buttons will create a PDF file of the inspections made during their inspection period. Once the inspector has generated any and all reports they desire, their use of the application is completed, and can exit the application.

6 References


7 Point of Contact

For further information regarding this document and project, please contact any member of Group 12. All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.